UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA ANIMAL

Construction of machine learning models to predict pharmacology properties of molecules

Rodrigo Daniel Garrilha Santos

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:
Prof. Doutor André Osório e Cruz de Azerêdo Falcão

2019

Resumo

O processo de desenvolvimento de drogas é altamente condicionado pela qualidade dos modelos com os quais se realiza a seleção dos primeiros compostos. Este trabalho procurou avaliar vários metodologias e descobrir qual a melhor abordagem para a construção de modelos de QSAR (relação quantitativa estrutura-propriedade/atividade) usando um conjunto grande de problemas.

Usando um banco de dados de modelação de problemas desenvolvidos no projeto de pesquisa MIMED, 500 conjunto de dados foram extraídos de forma a serem usados para a construção de modelos QSAR. Quarenta metodologias diferentes, resultantes na combinação de quatro algoritmos de machine learning, dois fingerprints e cinco valores de bits, foram usados para fazer os modelos. Com o uso destas metodologias forma criados 18000 modelos, dos quais após análise surgiu a abordagem que melhor generaliza os modelos. Esta é a combinação dos seguintes parâmetros: random forest without maximum depth com Extended-Connectivity Fingerprints de raio 2 usando 2048 bits. Esta abordagem após validação construiu modelos com valores RMSE (Root Mean Square Error) de 0.17 e valores PVE (Proportion of Variance Explained) de 0.63.

Por fim, procurou-se otimizar o processo de construção de modelos QSAR com a utilização da técnica de feature selection. Daqui resultou uma redução no conjunto de variáveis utilizadas pelo algoritmo resultando na construção de modelos mais robustos, mantendo o mesmo desempenho, RMSE de 0.17 e PVE de 0.59. Por fim a metodologia escolhida foi comparada com uma abordagem construída usando KNIME de forma a ter a perceção do fitness dos modelos construídos.

Keywords: Machine Learning, Chemioinformatics, Fingerprints, aprendização automática, seleção de variáveis, Python, SkLearn, QSAR

Abstract

The drug development process is highly conditioned by the quality of the mathematical models with which the first compounds are selected. In this work, we tried to evaluate various methods and find out which are the best parameters for building Quantitative Structure-Activity Relationship (QSAR) models using a large set of problems.

Using a database of modelling problems developed within the research project MIMED, 500 datasets were extracted to be used for building QSAR models. Forty different methodologies, resulting from the combination of four machine learning algorithms, two fingerprints and five bit values, were used to make the models. Using these methodologies, 18000 models were created, from which after analysis came the approach that best generalizes the models. This is the combination of the following parameters: random forest without maximum depth with Extended-Connectivity Fingerprints of radius 2 using 2048 bits. This approach after validation builds models with Root Mean Square Error (RMSE) values of 0.17 and Proportion of Variance Explained (PVE) values of 0.63.

After the choice of the methodology, we tried to optimize the process of building QSAR models using the feature selection technique. This resulted in a reduction in the set of variables used by the algorithm resulting in the construction of more robust models, maintaining the same performance, RMSE of 0.17 and PVE of 0.59. Finally, the chosen methodology was compared with an approach built using KNIME to have the perception of the fitness of the built models.

Keywords: Machine Learning, Chemioinformatics, Automatic Learning, Fingerprints, Selection of Variables, Python, SkLearn, QSAR

Resumo Alargado

O desenvolvimento de drogas novas é um processo muito complexo, demoroso, que consome muito dinheiro e que muitas vezes resulta no desenvolvimento de um produto que na reta final de aprovação falha a cumprir os requisitos necessários. Como tal, qualquer hipótese que haja para reduzir o tempo, o preço, a complexidade e aumentar o sucesso do processo é importante para o estudo. O desenvolvimento de drogas computacionalmente assistido apresenta-se como um método eficiente e capaz de contribuir para o auxílio e resolução destes problemas. Comparado às técnicas tradicionais de descoberta e desenvolvimento de novas drogas, o desenvolvimento de drogas computacionalmente assistido tira não só proveito de tecnologias referentes a aprendizagem automática como também de conhecimentos biológicos prévios e grandes quantidades de informação para uma seleção rápida e precisa de moléculas, com propriedades farmacológicas de interesse para o problema em estudo, para posterior análise em laboratório. Este desenvolvimento pode ser baseado em dois ramos, o design de drogas baseado nas estruturas (SBDD) e o design de drogas baseado nos ligandos (LBDD). A primeira etapa deste processo, denominada de descoberta de drogas, é responsável pela escolha de moléculas com propriedades farmacêuticas com potencial terapêutico para tratar uma determinada doença. Com o auxílio de métodos de aprendizagem automática, técnicas de data-mining e do uso da relação quantitativa estrutura-propriedade/atividade (QSAR), é possível construir modelos que consigam prever as propriedades que uma determinada molécula tem antes desta entrar em fase de testes laboratoriais. Tal deve-se ao uso de métodos QSAR para construir modelos capazes de prever as propriedades farmacológicas de uma molécula através da sua estrutura química. Durante a construção dos modelos QSAR existe alguns aspetos que se devem ter em conta, como

a curadoria dos conjuntos de dados, a escolha do tipo de algoritmo de aprendizagem automática a ser usado e a forma de representar as estruturas químicas dos conjuntos de dados (fingerprints). Este trabalho tem como objetivo a seleção da melhor metodologia, dentro de um conjunto de 40 abordagens, capaz de construir modelos QSAR a partir de um conjunto grande de problemas, que dê resultados consistentemente bons e robustos. Para tal, recorreu-se à base de dados do projeto MIMED (Mining the Molecular Metric Space for Drug Design), da qual se extraiu 500 conjuntos de dados que representam os problemas a serem modelados. Estes sofreram múltiplos processos de manipulação e transformação, através de um script desenvolvido, de forma a poderem ser usados para a criação de modelos preditivos. Primeiro , estes conjuntos sofreram um processo de limpeza, levando a uma redução do tamanho dos dados de cerca de 49%. De seguida, as atividades de cada molécula dentro do conjunto de dados sofreram uma normalização dos seus valores para um intervalo de 0 a 1. Esta normalização divide-se em dois passos: a passagem de todas as unidades das atividades registadas para nM e a consequente utilização de uma função logarítmica para transformar estas atividades no intervalo acima referido. Dentro de cada conjunto de dados existe um aglomerado de informação, não sendo toda ela necessária para o processo de criação de modelos. Tendo isto em conta apenas os campos contendo o ChEMBL ID, a atividade normalizada e o formato Simplified molecular-input (SMILES) de cada molécula foram guardados num ficheiro novo, sendo que estas três informações são as necessárias para criar modelos, ficando assim o novo ficheiro mais compreensível e limpo. Tendo os conjuntos de dados sido limpos, procede-se ao cálculo das fingerprints com o auxílio da biblioteca SkLearn. Estas são uma forma de representação da molécula em formato numérico, capaz de ser usado pelo algoritmo para construir os modelos de previsão. São usados dois tipos de fingerprints neste trabalho, o extended circular fingerprint de raio 2 (ECFP4) e o extended circular fingerprint de

raio 3 (ECFP6), em que a grande diferença reside no número de átomos que está a ser abrangido Procede-se à construção dos modelos QSAR. Cada conjunto de dados sofreu uma partição aleatória da sua informação através de uma função, em que 75% dos dados vão para o conjunto de treino, usado para treinar o modelo com o auxílio da técnica de N-Cross Fold Validation, e 25% para o conjunto de IVS, usado para validar o modelo. Estes conjuntos de treino e de IVS são usados por 40 abordagens distintas que se baseiam na combinação de três parâmetros: 4 algoritmos da aprendizagem automática, 2 fingerprints e 5 números de bits. Os modelos são avaliados consoante duas variáveis estatísticas, o Root Mean Square Error (RMSE) que indica a qualidade do modelo construído refletindo a diferença entre a "verdade" e o que o modelo previu (quanto menor o valor de RMSE melhor) e a Proporção de Variância Explicada (PVE) que indica o poder de previsão do modelo (quando maior o valor de PVE melhor). Foi estabelecido um limite para o qual todos os modelos têm de obedecer para serem considerados modelos preditivos, RMSE < 0.3 e PVE > 0.3. Os modelos que obtivessem valores acima (para o RMSE) ou abaixo (para o PVE) não eram considerados. Tendo o threshold em conta durante os primeiros testes, 50 conjuntos de dados provaram ser difíceis de modelar, pelo que foram retirados do conjunto a ser modelado ficando reduzido a 450.

O uso das 40 metodologias diferentes nos 450 alvos a modelar resultou na construção de 18000 modelos QSAR. Após a análise dos dados obtidos com recurso à realização do teste de ranking Friedman, que atribui ranks aos modelos criados, conclui-se que a melhor abordagem a ser usada dentro do leque de metodologias usadas, é a combinação dos parâmetros: algoritmo "random forest without maximum depth" acoplado aos ECFP4 com 2048 bits. O uso desta abordagem para os 450 problemas originou uma mediana para o RMSE de 0.17 e para o PVE de 0.63. Com a melhor abordagem selecionada procurou-se aumentar a performance da construção de modelos. Para tal, usou-se a técnica de "feature selection", que consiste na escolha das variáveis que

mais contribuem para a construção dos modelos, não usando para a construção do modelo as restantes. Ao utilizar este método aumenta-se a robustez dos modelos gerados, diminuindo-se a hipótese de "overfitting". Com o uso do "feature selection", corre-se o risco da perda de informação visto que se está a eliminar variáveis usadas na construção do modelo. No entanto, os resultados obtidos demonstram que tal não aconteceu uma vez que o RMSE obtido tem um valor de 0.17 e o PVE um valor de 0.59, muito próximos dos valores obtidos sem a realização do "feature selection". Finalmente realizou-se uma comparação direta da metodologia escolhida neste trabalho, com uma metodologia desenvolvida noutro sistema, nomeadamente, no KNIME. O objetivo deste passo é perceber como os resultados dos modelos desenvolvidos neste trabalho se comportam comparados com os resultados de modelos desenvolvidos usando outra abordagem. Para tal, dez alvos foram aleatoriamente escolhidos e usados por ambas as pipelines para construir os modelos usando os parâmetros escolhidos neste trabalho. Dez modelos foram feitos com a metodologia de eleição e vinte foram feitos usando a abordagem do KNIME, sendo que destes vinte , dez usam um tipo de fingerprints (Kausar_1) e os outros dez usam fingerprints diferentes (Kausar_2). Conclui-se que a abordagem Kausar_1 é a que produz modelos com melhores resultados, no entanto realizando o teste de Friedman não existe diferença estatística suficiente para afirmar que uma abordagem prevalece em relação às outras. É de notar que a metodologia escolhida neste trabalho, quando comparada com as duas abordagens usadas em KNIME, produz modelos num menor tempo. Futuramente, o ideal seria, com a abordagem selecionada, realizar um teste funcional para um problema em especifico, ou seja, construir modelos QSAR com a metodologia selecionada de modo a selecionar moleculas capazes de resolver o problema e testá-las *in vitro*. Seria também interessante experimentar novas técnicas de construção de fingerprints, e também algoritmos de aprendizagem automática de forma a englobar mais problemas, mantendo a robustez e performance já existentes.

Acknowledgements

First of all, I want to thank my grandmother Ibraíma Alves, for all you have done for me, you have been and you are a fundamental pillar in my life. Without you, this had never happened. A big kiss from the "Rasteirinho" who loves you so much.

I want to thank my girlfriend, Leonor Sousa Serranheira, for the unconditional support and love she gave me during this work. Thanks to you I was able to keep my head up high, never letting me give up even during the most difficult moments of this. You were undoubtedly my refuge during the storms that passed. Love you a lot "Picarrucha".

To my brother, Guilherme Garrilha, for helping me, in good times and bad. You were an essential part of my journey, as you showed me that no matter how complicated the path, the important thing is never to give up and keep trying, and you are a good example of that. Thank you for cheering me up when I was sadder. I love you brother.

I would also like to thank Professor Florentino Serranheira for his support and advice given to the realization of this work.

Last but not least, I would like to thank my advisor Professor André Falcão for all the support and guidance he gave me during this work.

Finally, I want to thank the Foundation for Science and Technology (FCT) for funding my Master's scholarship under the MIMED Project (PTDC / EEI-ESS / 4923/2014).

I want to dedicate this dissertation to my parents, José Marques dos Santos and Graça Maria Alves Pereira dos Santos.

Without you none of this would be possible, you did everything you could, gave me everything you had, never denying anything, doing your best to give me something you never could have. And for that, I am eternally grateful to you. I hope someday I can repay you for what you did and continued to do for me in some way. Thank you, Mom and Dad.

I love you both from the bottom of my heart.

Contents

1 Introduction **1**

1.1 Motivation . . . 4
1.2 Objectives . . . 4
1.3 Contributions . . . 4
1.4 Overview . . . 5

2 Background **7**

2.1 Computer Aided Drug Design . . . 7
2.1.1 Structure based drug design (SBDD) . . . 8
2.1.2 Ligand-based drug design (LBDD) . . . 9
2.1.2.1 Quantitative Structure-Activity Relationship (QSAR) 10
2.1.2.2 Pharmacophore modeling . . . 12
2.2 Digital Molecular Representation . . . 12
2.2.1 Molecular Descriptors . . . 13
2.2.2 Molecular fingerprints . . . 14
2.3 Supervised Machine Learning . . . 16
2.3.1 Linear Models . . . 18
2.3.2 Non-Linear Models . . . 19
2.3.2.1 Support Vector Machines . . . 19
2.3.2.2 Random Forests . . . 20
2.3.3 Workflows . . . 21

3 Data and Methods **23**

3.1 Data . . . 23
3.1.1 Data description . . . 23

3.1.2 Data cleaning . . . 25
3.1.3 Data Treatment . . . 25
3.2 Methods . . . 27
3.2.1 QSAR model Fitting . . . 28
3.2.1.1 Data partition and fingerprints calculation . . . 28
3.2.1.2 N-Fold Cross Validation . . . 29
3.2.1.3 Model Building . . . 29
3.2.1.4 Model Validation . . . 30
3.2.2 Model Ranking . . . 30
3.2.3 Modeling with Feature Selection . . . 32
3.2.4 Software . . . 32

4 Results **35**
4.1 Parameterization of QSAR models . . . 35
4.2 Modeling using Feature Selection . . . 43

5 Discussion **47**
5.1 Data set handling result . . . 47
5.2 Parameterization of QSAR models results . . . 48
5.3 Modeling using Feacture Selection results . . . 50
5.3.1 Comparison of models with and without feature selection . . . 51
5.4 Kausar Pipeline . . . 52

6 Conclusions **57**
6.1 Future Work . . . 58

References **61**

List of Figures

1.1 Relationship between the increasing in the number of new molecular entities (NME) approvels and the money spent in Research and Development (Berger *et al.*, 2013). 1
1.2 Drug development timeline (Phillip, 2018) 2

2.1 Process of the CADD Macalino *et al.* (2015). 8
2.2 QSAR Workflow (Tropsha, 2010). 11
2.3 Various algorithms and their characteristics (Gortari *et al.*, 2017). 14
2.4 Representation of the parameter radius in the process of making a fingerprint (Rogers & Hahn, 2010). 15

3.1 Transformation of the units of the activity values. 26
3.2 Operation used to normalize activity values that were in percentage. 27
3.3 Logarithmic equation used to normalize activity values. 27

4.1 Distribution of the mean RMSE for all 40 approaches. 40
4.2 Comparison of the distribution of the RMSE for ranks 1, 10, 20 and 40. 41
4.3 Friedman ranking test plot. 42
4.4 Correlation between the number of variables and RMSE after feature selection. 44
4.5 Correlation between the values of RMSE before and after feature selection. 45
4.6 Correlation between the values of PVE before and after feature selection. 46

LIST OF FIGURES

5.1 Relation between size of the data set with the value of RMSE IVS using ECFP4. 49
5.2 Relation between size of the data set with the value of PVE IVS using ECFP4. 50
5.3 Comparison of 10 data sets constructed using different methodologies in terms of RMSE. 55
5.4 Comparison of 10 data sets constructed using different methodologies in terms of PVE. 55

List of Tables

2.1 Various algorithms and their characteristics. 9

3.1 Different types of data set used in the study. 24

4.1 Data set differentiation by test group. 35
4.2 Number of models made from 450 datasets. 36
4.3 Median values of training RMSE and PVE per assay group. . . . 37
4.4 Median values of training RMSE and PVE per machine learning algorithm. 37
4.5 Median of training RMSE and PVE for Random Forest MD = None per number of bits. 38
4.6 Median of training RMSE and PVE for Support Vector Regression per number of bits. 38
4.7 Comparison of the results of the best models. 39
4.8 RMSE and PVE before and after using feature selection. 43

5.1 Results for RMSE and PVE for different approaches. 51
5.2 Decoding data set name to the corresponding gene name for all 10 data sets chosen. 52
5.3 Results for RMSE and PVE using the first methodology. 53
5.4 Results for RMSE and PVE using the second methodology. 54

Chapter 1

Introduction

Drug development is an extremely expensive and highly complex process, which results in being very prone to failure.

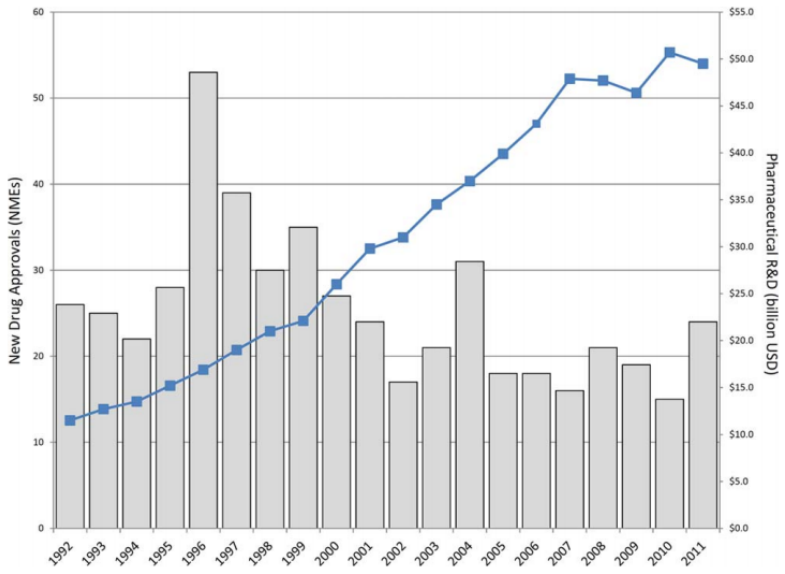

Figure 1.1: Relationship between the increasing in the number of new molecular entities (NME) approvels and the money spent in Research and Development (Berger *et al.*, 2013).

Multidisciplinary drug development teams have always been tasked with solving problems associated with this procedure (for example, the high amount of

money and resources spent) and have tried to find new methods that upgrade the quality of the drugs being developed as well as enhance the diagnoses of the problems.

This process is typically centered in 4 phases, from compound selection after knowing the nature of the problems to approval of the drug for human consumption.

As shown in the figure 1.2, these stages are called: Drug Discovery, Pre-Clinical development, Clinical development and Regulatory approval (FDA, 2019).

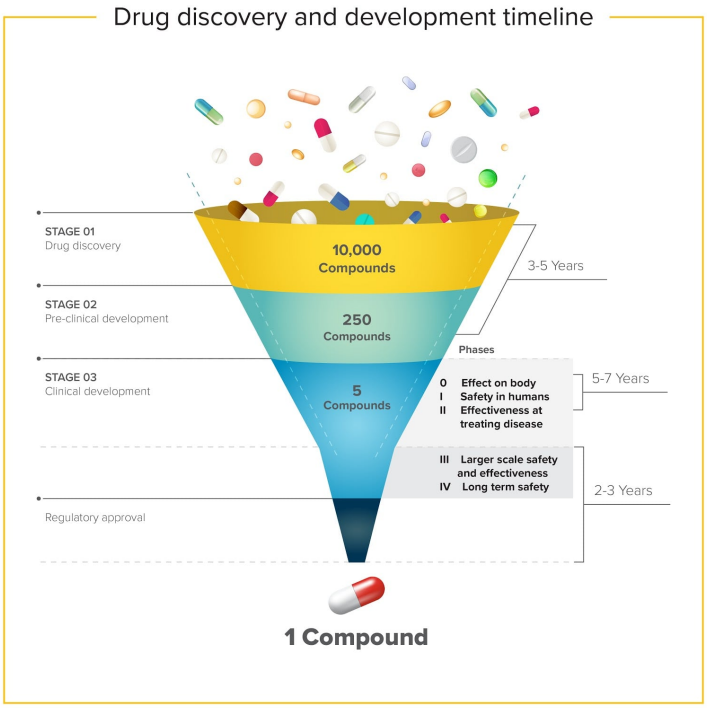

Figure 1.2: Drug development timeline (Phillip, 2018)

Drug Discovery: First, a disease is selected to be further investigated in order to find the mechanisms from which it's regulated. Through progress of fields such as biochemistry, new mechanisms of molecular regulation are discovered and with said discoveries, knowledge is obtained on how these diseases work. By these processes, molecular targets can be marked to be further studied and to identify their correlation with the disease and how said molecule influences the problem

when blocked or activated. Several laboratory tests are conducted in order to ensure that the elected targets are connected with the disease selected. Once validated, this step unlocks a scientist's ability to gather multiple chemical structures entities that probably have some therapy potential to treat the designated problem. After some short testing of the picked molecules, that list is shortened to a smaller number of compounds that looks promising. Subsequently this group of compounds is selected and more experiments are conducted in order to acquire information about them, with the intention of removing compounds who aren't fit for drug making.

Pre-Clinical Development: Tests *in vitro* and/or *in vivo* are made in order to obtain information about the compounds, mainly about the minimum dosing required for the compound take effect. the potency said compound, and furthermore to test the necessary dosage for it to be toxic to the patient. In this stage a large group of compounds are eliminated from the list of potential candidates to make a drug.

Clinical Development: *In vitro* and *in vivo* tests are capable of revealing a lot of information about the drug at issue, but it can't tell to the researchers how it will interact with the human body. To discover this particular issue, clinical tests are run in volunteers in order to understand how the drug reacts when put in direct contact with the human body. These tests are divided in 3 phases each and every one of them different, so researchers can ensure the drug developed is safe for human usage. Regarding the first phase, twenty to a hundred healthy candidates are hand-picked, to find out if the information, obtained in the pre-clinic tests, is either accurate or not (about 70% of tested drugs advance into the following stage). In the second phase, about 100 to 500 candidates with the disease, are introduced for the first time, to investigate the efficiency and the secondary effects of the drug (Only 33% of the tests performed in this phase progress to the succeeding one. The third and final phase is where the safety, efficiency and secondary reactions of the drug are monitored with 1000-5000 candidates that possess the disease (25-30% of the drugs tested in this phase advance to the next stage).

Regulatory Approval: If a drug passes all the tests, the company responsible for the development can fill an application to market the drug.

1.1 Motivation

The main incentive that led to producing this work was to find out if there's a consistently good method to return a robust model with good results regardless of the situation to model, while having a very large set of public data sets, based on two criteria, which are: being human proteins and also the size of the data sets being large enough.

1.2 Objectives

The main purpose of this dissertation is to test and evaluate if there's any methodology within a set of methodologies that's consistently better than the others to model a large set of problems in series.

Hypothesis: There is a methodology for parameterization and adjustment of machine learning models that can consistently yield good results regardless of the QSAR (Quantitative structure–activity relationship) problem to be considered.

1.3 Contributions

With the elaboration of this work, the main contributions resulting from this are the following:

Contribution 1: Selection of a methodology to build QSAR models using a big number of problems.

Contribution 2: Implementation of a QSAR method in Python with the help of scikit-learn.

Contribution 3: Set of parameters for building QSAR models with good results for a big number of targets;

1.4 Overview

The present document is organized by chapters containing six of them. The first chapter is a brief introduction to the problem as well as the history behind it, followed by this project's purpose and contributions to the scientific community.

The second chapter talks about the state of the art that will be used in this work, with more detailed information on the tools that are used, followed by the third chapter, the explanation of methods and data used to produce the results.

In the fourth chapter, the results obtained during the investigation are displayed and comments are made regarding their significance followed, by the fifth chapter, the discussion of the obtained results and what they mean to the problem.

Finally, in the sixth chapter, we spoke briefly about the outcome of the work done.

Chapter 2

Background

2.1 Computer Aided Drug Design

Like previously discussed, discovery of new molecules with pharmaceutical proprieties to known health problems is getting very expensive and time consuming. In order to tackle the problem computer aided drug discovery (CADD) was developed and nowadays more and more researchers tend to use this method, not only saving money, time and resources but because novel compounds can be tested *in silico* to verify their biological activity (Nantasenamat *et al.*, 2010). Doing this step can save a lot of money and time in later stages of drug design, where tests *in vivo* are needed and knowing the pre-result beforehand can be time and money saving (Camilo *et al.*, 2014). This method is preferred in comparation to alternative approaches like virtual high throughput screening (HTS) because, in contrary of other methods, CADD can predict new compounds that are biologically effective against a certain molecular target of study with higher success rates (Lionta *et al.*, 2014).

CADD have two different approaches, structure-based and ligand-based and one of those approaches needs to be chosen before doing the virtual screening of a chemical database in order to discover a new compound (Macalino *et al.*, 2015). Its also known that previous biological knowledge of the molecular target structure or of the ligands bioactivities is a major factor in the decision of which of both approaches should be chosen for the research.

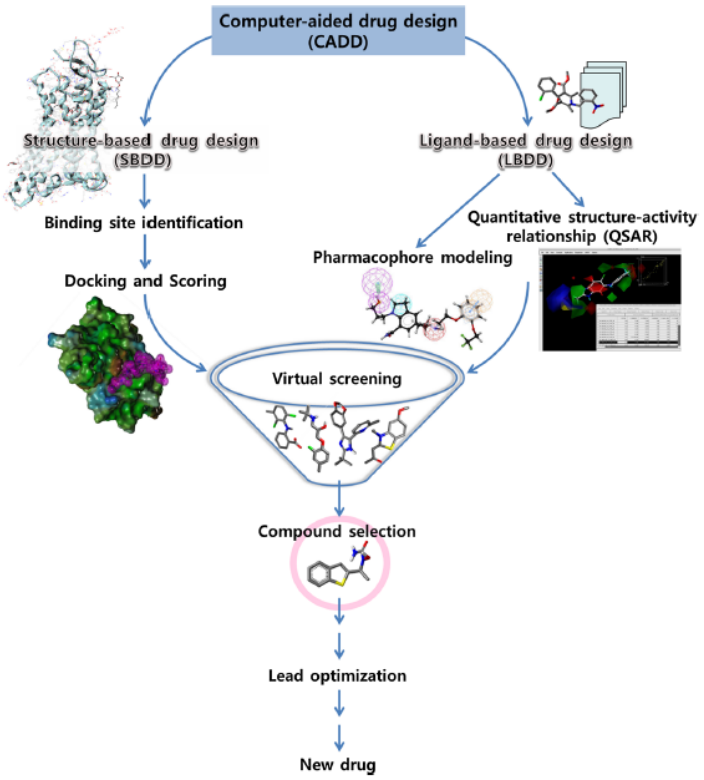

Figure 2.1: Process of the CADD Macalino *et al.* (2015).

2.1.1 Structure based drug design (SBDD)

SBDD is a method that uses ligand and target structures to perform structural analysis to deduce how the ligand interacts with the target. By using this approach, one can identify which ligand is best to inhibit/activate the target in question. According to Houston & Walkinshaw (2012) there are two major steps to perform a struture-based prediction: first is to sorting out which is the correct docking site for the ligand, in other words, the orientation and conformation that both the ligand and the target must have in order to interact. Second is to predict binding affinities close to experimental observations. The accuracy of the final score is dependent of the accuracy of the first step (Anderson, 2003).

Not all cases can use only one score algorithm like showed in Warren *et al.* (2006), As has been shown, the use of only one algorithm may result in certain cases in producing deceptive predictions about structural analysis between ligands

and targets. It is shown in Houston & Walkinshaw (2012) that combining at least two scoring algorithms, a method named consensus scoring, can increase the correctness of docked ligands up to 18%.

As seen so far docking is a crucial step to achieve success in SBDD, but its also a hard task to do, mainly because proteins don't have only one conformation, they have numerous conformations making the docking score process a difficult task to be performed. To make and explore the different conformations, SBDD uses software to predict the orientation of the ligand and the target (Grinter & Zou, 2014). Ligands are then treated like physical entities and different conformation tested, scored and ranked, through scoring functions with the objective to find the best position and orientation of the ligand, that binds with the target (Schneider & Böhm, 2002).

This models can be done, like said above, with a great variety of software available like, DOCK, AutoDOCK, LUDI, FlexX, GOLD and many others (Grinter & Zou, 2014; Schneider & Böhm, 2002).

Program	**Flexible Protein**	Flexible Ligand	Description
DOCK	No	Yes	Docks either small molecules or fragments, includes solvent effects.
FlexX	No	Yes	Incremental Construction
AUTODOCK	Yes	Yes	Uses averaged interaction energy grid to account for receptor conformations and simulated annealing for ligand conformations
LUDI	No	Yes	Docks and scores fragments

Table 2.1: Various algorithms and their characteristics.

2.1.2 Ligand-based drug design (LBDD)

LBDD is based in analysis of structural and/or activity, biological or chemical properties, data for compounds that have been tested in an assay against a biological target (Macalino *et al.*, 2015).

The logic behind this affirmation, is that using several active ligands of the molecular target it can infer not only its structure but also its chemical and/or molecular properties. It's done using the structural similarity of the molecules, which says that if molecules have similar structures, they will have similar biological activity. On this study it's used public data sets big enough to be modulated, containing ligands often used against a molecular target. Due to this fact, the LBDD category of CADD is the one that will be used.

This approach have a few techniques to find the new candidates, like the Pharmacophore modeling and the highly popular Quantitative Structure-Activity Relationship (QSAR).

2.1.2.1 Quantitative Structure-Activity Relationship (QSAR)

QSAR modeling is a technique that relies in the application of statistical or machine learning methods to predict the activity of compounds. It can be characterized by a collection of defined protocols and procedures that enable the application of this technique to explore collections of biologically active chemical compounds (Tropsha, 2010).

The goal of QSAR is to establish a connection between descriptor values and their biological activity. QSAR Modeling approaches imply, directly or indirectly, a simple similarity principle. The properties of a molecule are directly related to its structure, hence if the structure of a molecule is determined upon comparison with other similar compounds it's possible to infer its chemical properties (Tropsha, 2010). The objective of using QSAR is to generate statistical models able to predict biological properties of novel compounds (Macalino *et al.*, 2015) and to accomplish this a set of steps has to be performed.

The overall workflow can be described by the following steps (Kausar & Falcao, 2018; Tropsha, 2010):

- Curated the data set, guaranteeing the standardization of biological activity values. Performing this process ensures data quality for subsequent calculations;

- Proper splitting of data in the training set and independent validation score set (IVS). The training set is therefore divided through the N-Cross Validation technique into multiple training and test sets, during the modulation stage, being these sets used to develop the models and validate them at an early stage. IVS is later used once to externally validate the model;
- During the modeling phase, multiple QSAR techniques based on the combination of various descriptors and machine learning algorithms, are used to build predictive models. These are evaluated by statistics that can be analyzed to give information about the acceptability of the constructed model;
- External validation of models using the IVS set is a critical step in the QSAR method. It's used to assure that the results are unbiased, to have the perception of the quality of the prediction models built.

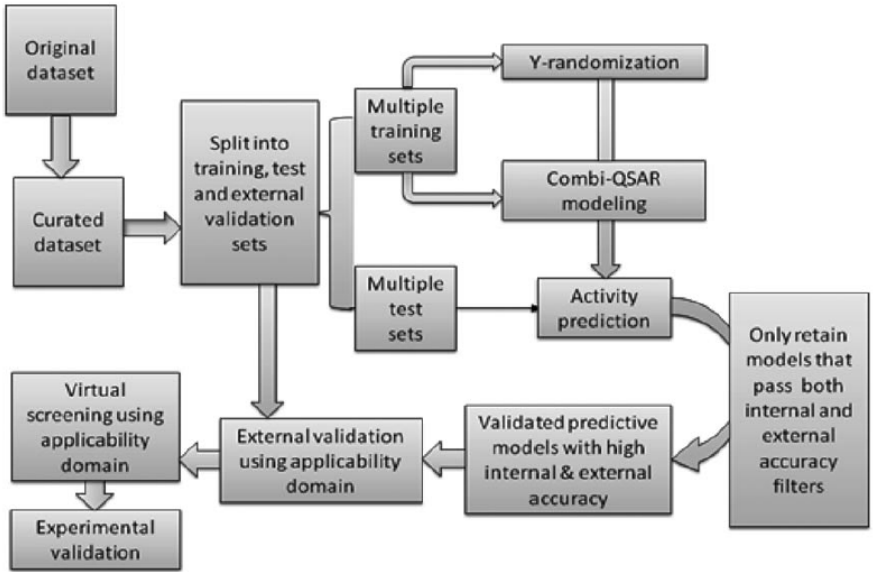

Figure 2.2: QSAR Workflow (Tropsha, 2010).

The ultimate goal of QSAR is to return a model with good predictive power, when applied to experimental validation, can return positive results in predicting the selection of chemical compounds (Tropsha, 2010).

There are a lot of different approaches for QSAR. Comparative Molecular Field Analysis (CoMFA) is a highly successful approach that is frequently employed. It uses the representation of ligand molecules, by their steric and electrostic fields, which by data analysis using cross-validation can predict the likelihood of having similar biological activities (ASIKAINEN *et al.*, 2005; Cramer *et al.*, 1988). Generating Optimal Linear PLS Estimations (GOLPE) is another approach that was born of the need of selecting the most important variables to making a more precise PLS model. The objetive is to find the best set of variables able to make the best predictive PLS model (Baroni *et al.*, 1993).

2.1.2.2 Pharmacophore modeling

The main goal for a pharmacophore model is to predict if a set of ligands are active or inactive for a given protein target, gives us as an output a list of active ligands. The concept of the pharmacophore model has been redefined over the last years and nowadays it can be defined as a model that describes the common properties that a group of ligands needs to have to bind to a specific receptor.

This can be verified when the 3D arrangements of molecules are similar, leading to the inference that molecules with similar 3D arrangements have similar biological activities/chemical properties. Following the latest reasoning, this approach has the principle that if multiple molecules bind to the same receptor to block/activate a protein, they will share similar chemical properties (Vuorinen & Schuster, 2015).

There are a different set of ligand-based tools available to anyone who wants to work with this type of model, for example, LigandScout, Discovery Studio, MOE, PHASE and many others (Vuorinen & Schuster, 2015).

2.2 Digital Molecular Representation

When representing molecules digitally, it is extremely important to use methods capable of differentiating similar compounds when they are compared, as the slightest difference between a pair of atoms can result in a large change in the chemical/biological properties of the molecule.

When representing molecules digitally two major concerns arise in relation to it, the representation and comparison of the molecules. Digitally representing molecules is a major challenge, mainly because they can have many different properties, structures, and sizes. It is important to realize that it is not always possible to transform the molecule as it is, leading to the loss of information during the transformation process.

Besides this, there is another problem, the choice of the method of representation of the molecule. There is a diverse range of representations available, for example, the IUPAC International Chemical Identifier (InChI) and the Simplified Molecular Input Line Entry System (SMILES). Even if the method of representation chosen is the most suitable, one still has to choose the method of comparison. Each case is different and because of this, there is no overall comparison method that works with all cases in general. Currently, there are two major groups where this comparison method can be fit: Molecular Descriptors and Molecular Fingerprint.

2.2.1 Molecular Descriptors

As discussed previously, to process the chemical information of a given molecule through a computerized way like QSAR, there is a need to represent this information in a format that is possible to quantify its characteristics. Molecular descriptors are a result of a logical and mathematical procedure where the representation of chemical information of a given molecule is transformed into numeric values represented by vectors (Grisoni *et al.*, 2018).

Depending on the type chosen for descriptors building, vectors can be differentiated and classified through dimensions. Nowadays exist 5 classes of molecular descriptors (0D, 1D, 2D, 3D, and 4D), each one more complex than the previous therefore containing more information about the molecule. The type of class chosen depends on the type of work to be developed because different problems holds different information about the target (Oprea, 2002). The first class (0D) contains information about simple characteristics like molecular weight, the second class (1D) represents fingerprints of the molecule or, in another words, fragments of the molecule, the third class (2D) represents the disposition of the molecule

in space in terms of atoms and their type, the fourth class (3D) represents the geometrical representation, the descriptor is created from the spatial (x, y, z) coordinates, of the 2D class, and finally, the fifth class (4D) represents the geometrical representation of the molecule (like the class 3D) but by introducing the flexibility factor, mimicking the ability of the molecule to change shape. (Grisoni *et al.*, 2018).

Classes of molecular descriptors aim to mimic the behavior of the molecules. This way researchers are able to understand the way it interacts with other molecules, their properties, and biological activities before actually doing the laboratory experiments.

The developed work uses 2D-molecular descriptors since they derived from algorithms applied to a topological representation.

2.2.2 Molecular fingerprints

Molecular fingerprints are a way of encoding the structure representation/chemical information of compounds. As well as molecular descriptors, molecular fingerprints are one of the most important parameters in virtual screening studies. This parameter is critical in predicting molecules in QSAR methods and the wrong choice of fingerprints can result in compromising the prediction model, causing the prediction results of the prediction model to be erroneous since the model has learned wrong information through the descriptors.

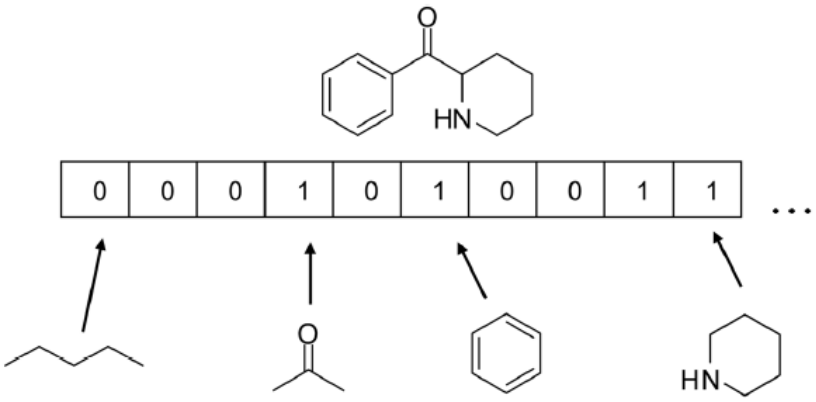

Figure 2.3: Various algorithms and their characteristics (Gortari *et al.*, 2017).

Fingerprints technique is built upon two characteristics, hashed and circular, and each add different information to the final result. Hashed is a bit string, that can have values between 0 and 1. When a bit is encoded with the value = 0 it means that there is in a given position of the molecule, an absence of a certain feature. On the contrary, when bit has the value = 1, it means that the feature is present in the position (Gortari *et al.*, 2017). The circular characteristic evaluates the neighborhood of each atom with a defined radius. The radius is the parameter that describes the distance to be evaluated, meaning that if a radius of 1 is chosen, all the adjacent atoms to the atom being evaluated are considered to the final score, if the radius is 2, all the atoms adjacent to the atom being evaluated are considered as well as all the atoms that have a bond with the atoms adjacent to the atom in study. The logic for other numbers of radius follows the same logic.

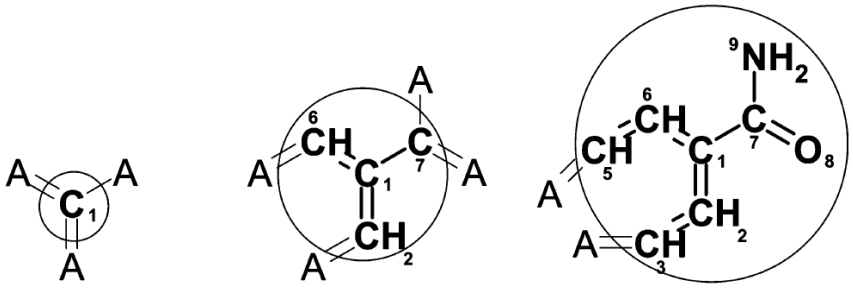

Figure 2.4: Representation of the parameter radius in the process of making a fingerprint (Rogers & Hahn, 2010).

A variant of the Morgan algorithm (Morgan, 1965) (a method used for identifying when two molecules, with different atom numberings, are the same), called Extended-Connectivity FingerPrints (ECFP) which is a combination of the Morgan algorithm with alternative methods of atom identifiers and performance techniques (Rogers & Hahn, 2010). ECFP is generated quickly, has a big range of applications, it has features very important for analyzing molecular activity and is a highly effective representation of the 2D molecular descriptors (topological representation of structural information).

When provided with a radius and a number of bits, it gives a bit vector with information about the chemical structure of a molecule. To approximate ECFP ,

the Morgan fingerprints were deployed from RDKit using default settings and an appropriate radius. Two versions were chosen for this study, the ECFP4, and the ECFP6. ECFP4, as explained used the Morgan fingerprint where the parameter that controls the number of iterations, the radius, is set with a value of 2 and ECFP6 the value used for the radius is set at 3. Both these ECFPs are tested in this methodology (Rogers & Hahn, 2010).

2.3 Supervised Machine Learning

Machine learning is the scientific study of algorithms and statistical models that perform specific tasks without the implicit indication of instructions. These models are built from a training set, from which the model distinguish the patterns and make inferences. In a final phase, the machine learning model receives a IVS, whose objective is to understand the quality of the model. There are two types of tasks: supervised and unsupervised, and with these different types of algorithms associated with them. Supervised machine learning uses function, like the one shown in equation 2.1, this method fits a given set of parameters, f, (differ consonant the algorithm used) that predicts a response variable, $\hat{y}$, from a feature vector, x.

$$\hat{y} = f(X) \tag{2.1}$$

It's really important to note that the output of the task always depends on the type of supervised machine learning that is used by the function. There are two types of supervised machine learning: Classification, where the response variable is discrete and Regression, where the response variable is continuous (numeric).

A supervised machine learning algorithm constructs a model using labeled observations from which it teaches the model how to analyze the data and afterwards, when the model receives a test set, it can assign each new entry to a class or predict a number[1]. Each supervised machine learning method has different

[1]https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/, last acceded on 30 May 2019

ways of evaluation. For example, the Classification method uses a confusion matrix to be evaluated and the Regression method uses the Root Mean Square Error (RMSE) (Schrider, 2018; Teixeira *et al.*, 2013) and the Proportion of Variance Explained (PVE) (Teixeira *et al.*, 2013). RMSE is a statistical variable that measures the observed difference between what values are predicted by a model and the actual values observed, allowing the qualification of the fit between the data and the model, showing the error of distribution (Kausar & Falcao, 2019; Teixeira *et al.*, 2013). RMSE is calculated using the module SKLearn. The RMSE formula is the following:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{2.2}$$

Where $\hat{y}_i$ is the estimation for the dependent variable y_i and N is the number of predictions. The best models will have low value of RMSE, meaning that what the model predicted in comparison with what the value is, is very close.

The PVE measures the proportion to which the model accounts for the variation of a given data set. For the perfect regression model, the percentage of variance explained is 100% when the model has 100% accuracy. When the PVE decreases it means that the estimation power of the model also decreases. For example, if the reaches zero the model has no predictive power. Just like RMSE, PVE is calculated using the SKLearn module. PVE is calculated using the following formula:

$$PVE = 1 - \frac{MSE}{VAR} \tag{2.3}$$

Where MSE stands for mean square error which is the difference between measured and predicted biological activity values, and VAR for variance, which is the difference between measured biological activity values and the average of all compound activities within the dataset.

There are other forms of evaluating the fitness of the model that was trained using supervised machine learning, but the most common are the ones described above.

Normally associated with the classification method there are algorithms like K Nearest Neighbour, Logistic Regression and Support Vector Machines. With the regression method, there are Linear regression, Regression Forest and Support Vector Machine - Regression.

2.3.1 Linear Models

Linear models use the technique of linear regression that is a modeling of the relationship between dependent variables and independent variables, assuming a linear relationship between them. In the current case of study, those are the molecular fingerprints and the biological activity of molecules respectively. The use of linear models has the objective of understanding if molecular fingerprints have a positive or negative effect on the characteristics of interest (Y) when constructing the predictive model. In other words, whether fingerprints contribute with new knowledge to model learning or not. It is called multiple linear regression because it isn't only one variable being modeled, instead, multiple variables are used in the modeling (Freedman, 2009).

Like previously said, linear regression is used for two types of situations, finding out if independent variables do a good job in predicting dependable variables and which variables are important predictors for predicting the dependable variables[1]. The simplest form of the regression equation can be defined with the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

In the previous equation each variable means the following:

Y $\rightarrow$ Characteristic of interest that we want to predict;

$X_{0,...,n}$ $\rightarrow$ Molecular Descriptor;

$\beta_{0,...,n}$ $\rightarrow$ Weight of each molecular descriptor;

[1]https://www.statisticssolutions.com/what-is-linear-regression/, last acceded on 31 May 2019

2.3.2 Non-Linear Models

Nonlinear models allow the adjustment of more complex relationships than linear or linearizable ones, between the dependent and independent variables, which in this case are the molecular fingerprints and the biological activity of molecules. In many cases, such models have their specific functional form for the problem being treated, because unlike a linear model, non-linear ones don't have a standard form. Non-linear models are characterized by the fact that the prediction equation depends non-linearly on one or more unknown parameters.

A non-linear regression models has the following form:

$$Y_i = f(x_i, \theta) + \epsilon_i, i = 1, ..., n \tag{2.4}$$

The Y_i are responses, f is a known function of the co-variate vector of predictor variables $x_i = (x_1,...,x_x)^T$ and the parameter vector $\vartheta = (\vartheta 1,...,x\vartheta p)^T$, and ε_i are random errors. The ε_i are usually assumed to be uncorrelated with mean zero and constant variance (Smyth, 2002).

Using machine learning there is an extensive library of non-linear algorithms that can be used to construct statistical models. Among many others, two of the most used algorithms are Random Forest (RF) and Support Vector Machine (SVM), due to the advantages they bring to model building (Kausar & Falcao, 2019, 2018; Teixeira *et al.*, 2013).

2.3.2.1 Support Vector Machines

Support Vector Machines (SVM) can address classification and regression problems. Given a train set, containing labeled data, an SVM training algorithm builds a model that assigns new data to the corresponding label, through the recognition of subtle patterns. An SVM model is a representation of data as points in space, mapped in a very high dimensional feature space or hyperplane so that data of different categories are divided by a clear gap that is as wide as possible. New data is later mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. Support Vector Regression (SVR) is one of two main categories of SVM, being the other Suppor Vector Classification (SVC), whose main feature is to minimize the generalized

error bound associated with model building, improving the overall performance (Basak & Patranabis, 2007). In non-linear modeling is used the Kernel function. Two types of kernels could be used in our work: linear and radial basis functional (RBF). The difference between both is that RBF kernels can nonlinearly map samples into a higher dimensional space and linear cannot. Given the nature of our problem where the relationship between labels and attributes is nonlinear, using SVR with RBF kernels is preferred.

2.3.2.2 Random Forests

Random Forests (RF) is an ensemble method for classification and regression when joined with bootstrapping creates a powerful algorithm to generate machine learning models. RF is an algorithm that is made out of multiple decision trees, a forest, and each decision tree is made by bootstrapping. Bootstrapping (Efron, 1979) is a statistical re-sampling technique that involves random sampling of a dataset with replacement. It is often used as a means of quantifying the uncertainty associated with a machine learning model. In other words, RF creates decision trees using bootstrapping, being each tree constructed independently of previous trees using a different bootstrap sample of data, that work over a random subset of the data set, selecting it arbitrarily at each node, been each one chosen by their ability to divide the sample, building a large collection of correlated trees.

After several independent weak trees (forest) have been created, the RF algorithm uses a consensus voting process to combine the independent forecasts generated through the weak trees to generate a model that can make optimized predictions more robustly (Hastie, 2008; Teixeira *et al.*, 2013). The generalization of this method depends on the strength of the individual trees in the forest and the correlation between them. In terms of prediction accuracy is one of the top performs, that provide unique features for QSAR models building. Among them, one of the most important that this ML algorithm can measure descriptor importance as well as the similarity between molecules (Genuer *et al.*, 2010; Svetnik *et al.*, 2003).

An RF is made up of several parameters, and in this work, the "maximum depth" parameter (max_depth) had a higher relevance compared to the others, that were left with the "default" values (Teixeira *et al.*, 2013). Max_depth represents the depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data.

2.3.3 Workflows

The use of QSAR models for the construction of predictive models of molecules with pharmacological properties of interest is not new. There is a huge range in the example literature where QSAR methodologies are implemented to accomplish this task.

A large portion of the literature is focused on the development and implementation of a pipeline for a specific target, obtaining good models for it (Martins *et al.*, 2012). There are works, however, where the aim is to compare the effect of using different parameters within a small set of targets (Kausar & Falcao, 2019, 2018), aiming to improve the performance of the approach used to build better and more robust fitness models (Teixeira *et al.*, 2013). Others try to simplify and automate the process of building QSAR models by allowing the use of QSAR techniques by the scientific community that has little programming knowledge (Cox *et al.*, 2013; Dixon *et al.*, 2013).

However, these approaches have something in common. If one tried to modulate a large set of targets, as is the case under study, these methodologies would not be the most appropriate to do so. This is because they take a long time to model a problem, in other words, these approaches involve a very large computational cost for large data sets (Kausar & Falcao, 2019).

Chapter 3

Data and Methods

3.1 Data

3.1.1 Data description

The datasets used in the study comes from the database of the project MIMED (Abrantes, 2014). A pipeline was developed to fetch molecules with therapeutically potential from online repositories, namely ChEMBL (Gaulton *et al.*, 2017), and gathering them in a database where users could download the molecules to data sets. Data undergo multiple steps to be used, such as the manipulation of the raw information retrieved from the databases to produce data containing results to be used in the future (Abrantes, 2014). From this work, 500 datasets containing information about different chemical tests and activity measures were extracted. As shown below in table 3.1 the data sets can be divided into 3 big assay groups A, B and F, each meaning respectively ADME (Absorption, Distribution, Metabolism, and Excretion), Binding and Functional Assays. Inside each group, it's found different activity measures: IC50, EC50, Potency, Ki, Kd, Inhibition, AC50, RBA, and Activity:

- EC50: Half Maximal inhibitory concentration indicates the value required for a given substance to inhibit a given biological function.
- IC50: Half Maximal effective concentration represents the concentration required of a given compound to obtain 50% of the maximum effective

effect.

- Ki: The inhibitor constant, Ki, is an indication of how potent an inhibitor is; it is the concentration required to produce half-maximum inhibition.
- Kd: Binding affinity is typically measured and reported by the equilibrium dissociation constant (KD), which is used to evaluate and rank order strengths of biomolecular interactions. The smaller the KD value, the greater the binding affinity of the ligand for its target.
- Potency: Concentration required to elicit a specific response.
- Inhibition: Is the inhibition % under the assay conditions
- AC50: Concentration required for 50% activity;
- RBA: Relative binding affinity;
- Activity: Captures a range of non-standard activities described in references;

Activity Type	A	B	F	Total
IC50	5	191	15	221
EC50	0	4	12	16
Potency	3	0	10	13
Ki	0	93	26	119
Kd	0	2	0	2
Inhibition	5	80	0	85
AC50	0	1	1	2
RBA	0	1	0	1
Activity	1	0	0	1
Total	14	372	64	450

Table 3.1: Different types of data set used in the study.

3.1.2 Data cleaning

From MIMED Project's database, 500 data sets were extracted to be used and worked through all the workflow delimited at the beginning of the project. The retrieved information about the targets is far from being normalized, due to assay groups being different as well as the type of activity that each data set measures and how it does it. In a simple way, the data sets needed to be normalized due to their nature. This normalization occurs in two steps. The first step is to iterate over all data sets searching for missing or duplicate data inside them. The missing data would incapacitate our script from working and the duplicate would only over-fit the models not adding variety to them, making the models built with little robustness. When these criteria are met, the line in question is eliminated from the data set and the next in succession is tested for the same problems. This process was repeated through all lines inside the data set and after it through all data sets. Doing this results in the shortening of all data sets, mostly as a result of deleting lines that had incomplete or missing information about the molecules. After the process is completed, another step is activated to create a new data set file, SAR (Structure-Activity Relationship) file, containing only the necessary information to make models.

A SAR file has 3 columns, each one containing one type of data. The first one has the ChEMBL ID used to identify a molecule as well as it's properties, the second has the activity values for the molecule and the third column possesses the SMILES format set for retrieving the molecular structure of the molecule. With this information, QSAR models can be built.

The process resulted in 500 new SAR files containing only the necessary information from our 500 data sets. Before the cleaning process, all information gathered in the 500 data sets had a total size of 122 megabytes being reduced, after cleaning, to a total size of 62 megabytes.

3.1.3 Data Treatment

Following the previous reasoning, all the data used in the study had to undergo a treatment to be used. The objective of normalizing data is to reduce redun-

dancy, improve data integrity and facilitate data analyzing. This is done using an algorithm that processes information contained in our SAR file.

The first step in the process was to find missing data and duplicates of already existing data in our data and eliminate it. Doing so the data redundancy is reduced as well as missing information in some of the rows in the data sets, meaning a cleaner and simpler file to be handled to the model building algorithm, making the data set better to be worked with. After the first step, there could be two options, the first one is the case where the data is already normalized and ready to be worked, simply handled to the script. The second case, and more common, is where data needs to suffer a transformation through a logarithmic function (figure 3.3). In the current study case, the information contained on the data sets wasn't normalized, so the logarithmic function needs to be applied to all the 500 data sets in the study.

The way this block of code works is simple. It checks row by row, each line being a ligand of study, for its activity value, verifying the current quantity unit in which the activity was recorded. The activity was recorded using different units, nM, µM, mM, and percentage. Most of the data were registered using nM. Due to this and the need to normalize all recorded activities, all units that were not in nM were converted to this except for the recorded activities as a percentage (3.1).

$$\mu M * 1,000 = nM$$

$$mM * 1,000,000 = nM$$

Figure 3.1: Transformation of the units of the activity values.

For the activity to be used by the main algorithm to build models, its values had to be between 0 and 1. If the value registered on the activity was a percentage, only a simple division by 100 was needed to have the new value for the normalized activity (3.2).

$$New\_Activity = \frac{Activity}{100}$$

Figure 3.2: Operation used to normalize activity values that were in percentage.

The rest of the values, who undergo a unit transformation, most now be in nM, meaning that the logarithm equation can be used to normalize the values, fitting them in a range between 0 and 1. This transformation can occur in three different ways:

- In the case of the activity < 1, the new activity (spAct) registered will be 1 meaning that the quantity needed for the ligand, referenced in the row, to stimulate/inhibit is a target is very low;
- In the case of the activity > 10000, the new activity will be 0 being the logic equal to the previous point but in reverse, it's needed a great quantity of the ligand to stimulate/inhibit its target;
- In the case of the values of the activity is between 0 and 10000 nM, which is the majority of the cases, the logarithmic function is applied to retrieve a value between the range of 0 and 1. The scale follows the previous logic, values near to 1 mean less quantity needed and values closer to 0 means more quantity needed.

$$New\_Activity = spAct = \frac{4 - \log_{10}(Activity)}{4}$$

Figure 3.3: Logarithmic equation used to normalize activity values.

With the activity normalized, the SAR files are now fully operational to be used by the algorithm to build QSAR models.

3.2 Methods

This section will be described what software was used to perform the modeling as well as all the methods applied to obtain the results.

3.2.1 QSAR model Fitting

QSAR Models can be made using different approaches as previously stated. The objective of finding the best set of parameters capable of building the best model for a high number of targets isn't an easy task. Choosing the ideal bit number (128, 256, 512, 1024 or 2048), the fitting molecular descriptor and the best machine learning algorithm are some of the main concerns in terms of QSAR modeling.

Making QSAR models can be divided into three parts: calculation of molecular descriptors and partition of the data; build of the first models and selection of the "best" one; validation of the latest with the IVS set. To build a QSAR model it's necessary to follow the order described above or otherwise, the models resulting from this process will not be accurate enough to predict any chemical property. The developed algorithm works through layers of iterations. It goes by data sets, radius and finally bit numbers. Each iteration it fetches a different dataset, previously treated and transformed in a SAR file, selects a radius, a bit number and goes through the process described above, only changing to a different dataset after going through all the bits and radius combinations, repeating it all over again for all the datasets.

3.2.1.1 Data partition and fingerprints calculation

To do the first part, calculating the Morgan Fingerprints and partition of the data, the information inside the data sets needs to be extracted to be worked. This information will originate two variables, one with all the activities from the ligands and the other with the correspondent Morgan fingerprints. The Morgan fingerprints are calculated through a function when given the SMILES string of the ligand. Both variables are going to be divided thus creating two new sets of data, the Train set, and the Independent Validation Set (IVS), having 75% and 25% of all data correspondingly. The Train set will be used on the second step, to train our models, and the IVS set will be used on the third and final step to test and validate the model with the best parameters.

The objective of splitting data into training and testing sets is to maximize the best learning result and the best validation for the model. More data is allocated

to the training set because if the model doesn't learn it can't predict. Given the nature of the splitting, random split of the data, it's hard to reproduce directly the results obtained for both the train and the test stages.

3.2.1.2 N-Fold Cross Validation

To overcome this problem, the N-fold cross-validation method is used. Taking the Train set, containing 75% of the data, it's divided into "N" observations/partitions. From this "N" folds, one fold out of these "N" folds is chosen as testing data set (test fold) while the rest "N-1" is used for training the model (training folds), equal to the partition of the data in the initial step. Training sets are given to the model to learn from them. Then the test set is presented to the model, generating two statistical variables, Root-Mean-Square Error (RMSE) and Percentage of variance explained (PVE), that will be used to evaluate the fitness of the models. This procedure is repeated "N" times, each time selecting a different fold for the test set and the other folds (N-1) as training sets. Consequently, "N" different RMSE and PVE are produced from the procedure. With these, a mean is calculated to describe the overall results, higher the "N" the smaller the variance, of the learning capability of the model given the original train set (Koul *et al.*, 2018).

The objective of N-fold Cross Validation is to forecast how the models originated in this step will perform when exposed to the IVS, previewing which one will have the best fitness in the later stage of the process. This is done using two statistical variables RMSE and PVE. Given the values presented by them, the models can be evaluated in terms of learning and predicting. The criteria for choosing the best model is to sort all models by the lowest RMSE and the highest PVE.

3.2.1.3 Model Building

The models are made using 2 different machine learning (ML) algorithms, Random Forests (RF) and Support Vector Machine (SVM).

For RF three versions are tested, where what changes is the "maximum depth" parameter. In these approaches, three different values for maximum depth will

be used: 2,3 and None. The first two are used to obtain results and thus information that when increased the maximum depth better results are obtained, while the "true value" for this parameter is going to be "None", meaning that the algorithm won't have any restriction regarding the depths of each tree in the forest (Landrum, 2019)[1].

Combining machine learning algorithms with the N-Fold Cross-Validation technique originates multiple models. Of these, only the best is returned to be validated by IVS. The choice is made taking into account the RMSE and PVE values of each model, these values are compared by model, where the model with better fitness is chosen as the best model to model the problem in question.

3.2.1.4 Model Validation

Afterward, the model presenting the best combination of RMSE and PVE is selected to be presented to the IVS. It's worth noting that by radius is chosen, 20 different models are built, but only 1 is selected to be validated by the IVS. It's also to be noted that the results of the IVS are always worst than the Train set, since the model train with the Train set and when given the IVS, the data was never seen before during the training period. During this procedure all information about all models is being saved in two different CSV files to be analyzed at the end of it, to find the best set of parameters to make QSAR models. One CSV file will be saving all information about the models build in the N-fold cross-validation and the other file will only contain the information about the best model build for each SAR file.

3.2.2 Model Ranking

To get a more in-depth view of the results, a statistical test called Friedman ranked test was performed (Hollander *et al.*, 2015). This test is a non-parametric statistical test, or by other words is a test that used data that doesn't fit a well-understood distribution, that allows groupings of statistically indistinct treatments under the same grouping, meaning it captures differences between groups.

[1]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, last acceded on 24 June 2019

In the current work, the modeling approaches correspond to a treatment which is evaluated by its results for different data sets. Each model is ranked between the pool of models according to their RMSE scores. Having a lower rank means having a better performance and a higher rank means the worst performance. The Friedman test is then applied to understand if the different treatments are statistically different from each other to be considered better or worst (Hollander *et al.*, 2015; Kausar & Falcao, 2019). The process begins with the definition of two hypotheses to be proved:

- H_0: There isn't differences between "K" methodologies;
- H_1: There is differences between "K" methodologies;

Being "K" the number of methodologies tested.

After it, the alpha number is selected, being alpha the probability of H_0 being rejected, and the degrees of freedom calculated from a simple equation (k-1). Both variables are needed to find the value "X" for Chi-Square in the Chi-Square distribution table, from which the decision rule is made. This rule says that if the value calculated for Chi-Square is greater than "X" the null hypothesis is rejected and vice-versa. With this, the following equation 3.1 to calculate Chi-Square can be applied to find the exact value for Chi-Square for finding if the null hypothesis is rejected or not.

$$\chi^2 = \frac{12}{nk(k-1)} \sum R^2 - 3n(k+1) \tag{3.1}$$

In this work, "n" is the number of data sets used, "k" the number of different methodologies used and "R" is the sum of the rankings of the models.

In this case of study the treatments are the difference parameters, for example the set ML algorithm = SVR | radius = 2 | bits = 512 is one treatment and ML algorithm = SVR | radius = 3 | bits = 1024 is another. With this test ranking, the different sets of parameters for the dataset being modeled are possible, allowing to understand the differences between the treatments with a post hoc analysis.

The purpose of performing this test is to confirm if the conclusion made by analyzing both RMSE and PVE of the models, about what is the best approach

to build QSAR models does or doesn't have some statistical significance to be affirmed as the best set.

3.2.3 Modeling with Feature Selection

Even though data has been cleaned and redundant information deleted most of the time models are build with more variables than needed causing poor generalization of the information. The problem of selecting the minimal subset of descriptors/variables that can predict a certain pharmacology property with good performance, efficiency and in a robust way, can be solved using the technique of feature selection (Teixeira *et al.*, 2013). Using N-fold cross-validation and the ML algorithm random forest, the importance score of each variable is calculated to find variables highly related with the target (Genuer *et al.*, 2010), and sorted by the most relevant variables first and the less important ones last, through variable importance threshold. After this step, the variables are feed, step-wise, to an SVR model following the rank order (Genuer *et al.*, 2008).

3.2.4 Software

The main software used on this project was Python[1]. Python is an easy to learn language, full of packages and modules that helps interpret many situations and provides a lot of help when it comes to modeling data sets and analyzes the data statistically afterwords.

When it comes to modules used in the work, the main one is Anaconda[2]. Anaconda is a free and open-source distribution for both Python and R[3] with the purpose of doing scientific computing when it comes to deploying AI and machine learning. To complement this module, a library was added named SciKit-Learn[4], bringing with it a large selection of modules. From SKLearn, a module for the partitioning of the data set in the training set and the IVS, a module for calculating RMSE and PVE, and two modules for calling machine learning algorithms were used in the development of this work.

[1]https://www.python.org/doc/essays/blurb/, last acceded on 6 June 2019
[2]https://www.anaconda.com/what-is-anaconda/, last acceded 6 June 2019
[3]https://www.r-project.org/, last acceded on 6 June 2019
[4]https://scikit-learn.org/stable/, last acceded 6 June of 2019

Was also used RDKIT[1] which is an open-source toolkit for cheminformatics that as the power of processing the data inside of the datasets gives us the ability to work it. RDKIT has the power of calculating molecular descriptors, reading sets of molecules, calculate substructure of molecules, write sets of molecules between many other functionalities.

Lastly, KNIME (Berthold *et al.*, 2007) was used to build QSAR modeling using a different pipeline than the one used in this work. KNIME is a free and open-source data analytics, with the ability of integrate various components for machine learning and data mining through its modular data pipelining concept. By using it, its possible to generate results in a very efficient way through an automated pipeline.

[1] https://www.rdkit.org/docs/Overview.html, last acceded 6 June 2019

Chapter 4

Results

4.1 Parameterization of QSAR models

The originated models are evaluated according to a set of criteria, that when it fails to satisfy these criteria, it means that the model has lower or no prevision power. During the firsts runs of the script, many data sets kept giving problems due to their nature, meaning that they are a difficult problem to model and, therefore given this problem and the incapability of solving it, those 50 data sets were withdrawn from the poll of data sets. Doing this resulted in a total of 450 data sets for further work, with a total size of 52 Mb (table 4.1).

Assay Group	**Number of Data sets**	**Percentage**
A	14	3
F	64	14
B	372	83
Total	450	100

Table 4.1: Data set differentiation by test group.

With the new data sets normalized, tests were ready to be conducted by the algorithm, who is now able to run and originate forecasting models, that will be analyzed and compared between themselves. The algorithm has the configuration to run and build models within the range of all possible combinations of radius

of Morgan Fingerprints (2 and 3), number of bits (128, 256, 512, 1024 and 2048) and machine learning algorithms (Support Vector Regression, Random Forest with max. depth = 2, Random Forest with maximum depth = 3 and Random Forest without any maximum depth (= NONE)). The number of models builds by type can be seen in table 4.2.

Types	Models	Datasets
IC50	8440	211
EC50	640	16
Potency	520	13
Ki	4760	119
KD	80	2
Inhibition	3400	85
AC50	80	2
RBA	40	1
Activity	40	1
Total	18000	450

Table 4.2: Number of models made from 450 datasets.

The combination of all parameters resulted in 40 models created by data set. To find out the best parameterization to use for the creation of models for a big set of problems, the median for the RMSE and the PVE was calculated from the 18000 models originated from each methodology. The results of the combination can be seen in the following tables.

In table 4.3 the results correspond to an overview aspect of our data per assay group, giving us the perception of the quality of our data and what type of predictive models are being made by the algorithm.

Assay Group	ECFP4		ECFP6	
	Median RMSE Train	Median PVE Train	Median RMSE Train	Median PVE Train
A	0.163	0.219	0.165	0.188
B	0.209	0.414	0.219	0.357
F	0.16	0.306	0.164	0.259

Table 4.3: Median values of training RMSE and PVE per assay group.

Generally for each assay group, in Morgan Fingerprints radius = 2 (ECFP4) methodologies models tend to be better in both RMSE and PVE when compared to methodologies where Morgan Fingerprints are = 3 radius (ECFP6). Between each assay group through the analysis of the obtained values, the best models come from group B followed by group F and finally group A.

In table 4.4 it's notable the contribution of each machine learning algorithm for the making of the prediction models and which are better.

Machine Learning Algorithm	ECFP4		ECFP6	
	Median RMSE Train	Median PVE Train	Median RMSE Train	Median PVE Train
RF MD 2	0.2301	0.2724	0.2309	0.2682
RF MD 3	0.2151	0.3595	0.2157	0.3526
RF MD NONE	0.1769	0.5702	0.1796	0.5572
SVR	0.2062	0.4286	0.2316	0.2709

Table 4.4: Median values of training RMSE and PVE per machine learning algorithm.

The first notable thing about these results is the significant improvement in models created with RF as forest depth increases. When comparing SVR results and RF, the methodology that uses RF without depth limitation is the one that produces the best models with both ECFP4 and ECFP6.

All the models created with RF with maximum depth 2 and 3 were done with two purposes, of comparison with the model created with SVM and RF with a

maximum depth of none and to affirm that when RF models are built without limitation of the forest of trees, models have a better prediction power with lower error. Because of this only the results of the Random Forest without maximum depth and Support Vector Regression have been taken into account for the rest of the work leaving the other two aside.

Random Forest MD = NONE	ECFP4		ECFP6	
	Median RMSE Train	Median PVE Train	Median RMSE Train	Median PVE Train
128	0.1837	0.5421	0.1892	0.5218
256	0.1788	0.5668	0.1816	0.5485
512	0.1758	0.5753	0.1781	0.5645
1024	0.1749	0.5814	0.175	0.5771
2048	0.1726	0.5891	0.1736	0.5795

Table 4.5: Median of training RMSE and PVE for Random Forest MD = None per number of bits.

Looking at the overall results it can be seen an increase in performance as the number of bits goes from 128 up to 2048, also models made with ECFP4 seem to have a little better performance than models made with ECFP6, as seen before. Between the number of bits 1024 and 2048 both RMSE and PVE, for both ECFP4 and ECFP6, are almost identical being the models made by 2048 slightly better (table 4.5).

Support Vector Regression	ECFP4		ECFP6	
	Median RMSE Train	Median PVE Train	Median RMSE Train	Median PVE Train
128	0.1941	0.4841	0.2074	0.4238
256	0.2026	0.4524	0.2256	0.3282
512	0.209	0.4271	0.2371	0.2662
1024	0.2121	0.4021	0.2419	0.2372
2048	0.2139	0.3969	0.2447	0.2232

Table 4.6: Median of training RMSE and PVE for Support Vector Regression per number of bits.

For the models made with SVR, there is also a difference in the results as the number of bits rises. In contrary to random forest models, as the number of bits rises the results decreases, being the best models the ones made with 128 bits with ECFP4 (table 4.6).

Analyzing the best models made out of the 450 data sets, from the 900 QSAR models (450 for each ECFP) made, 898 were build using RF without maximum depth. In terms of ECFP, fingerprints made with radius = 2 have slightly better results compared with models made with radius = 3. The same can be said for the bit numbers, models made with 2048 bits have a little bit better results compared with models made with both 512 and 1024 bits (table 4.7).

Bit Number	ECFP4				ECFP6			
	Median RMSE Train	Median PVE Train	Median RMSE IVS	Median PVE IVS	Median RMSE Train	Median PVE Train	Median RMSE IVS	Median PVE IVS
512	0.1765	0.5839	0.1832	0.5459	0.1781	0.5714	0.1843	0.5575
1024	0.1631	0.6299	0.1682	0.6076	0.1724	0.6082	0.1752	0.5969
2048	0.1682	0.6413	0.1711	0.6366	0.1698	0.6243	0.1706	0.6132

Table 4.7: Comparison of the results of the best models.

To complete the results presented in the tables, three plots were performed through the 40 tested approaches, comparing the average RMSE obtained by each one.

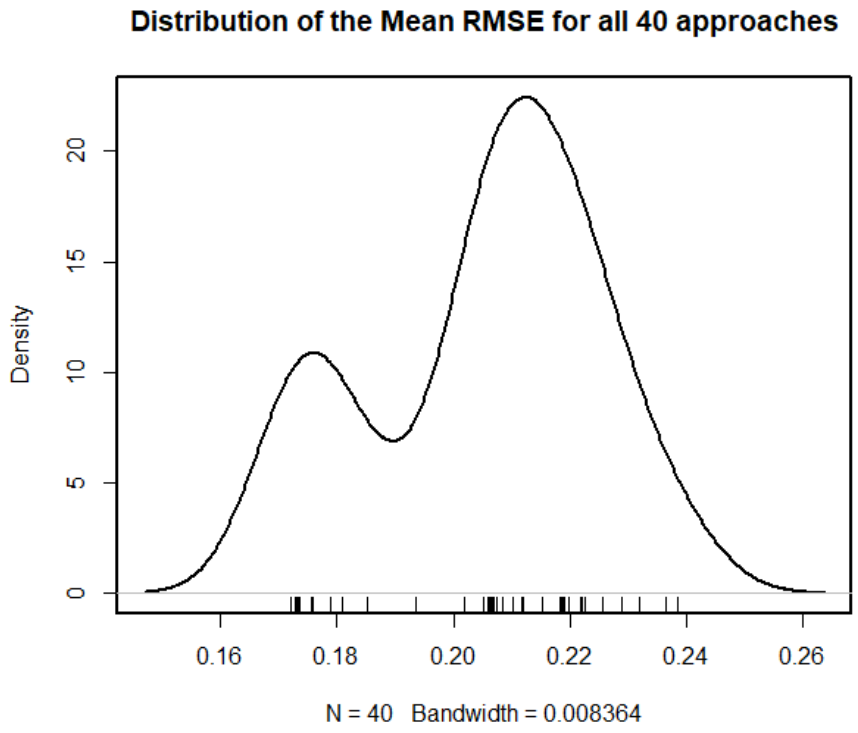

Figure 4.1: Distribution of the mean RMSE for all 40 approaches.

The above density plot (figure 4.1) compares the mean RMSE values for all 40 approaches used and how it is distributed over its range of values. Notable on the plot is the existence of two large groups. The first group (leftmost), shows a central tendency (mean RMSE) close to 0.17 while the second group (rightmost), has a central tendency between 0.21 and 0.22. This indicates that there is undoubtedly a group of methodologies which in comparison with the others behave better.

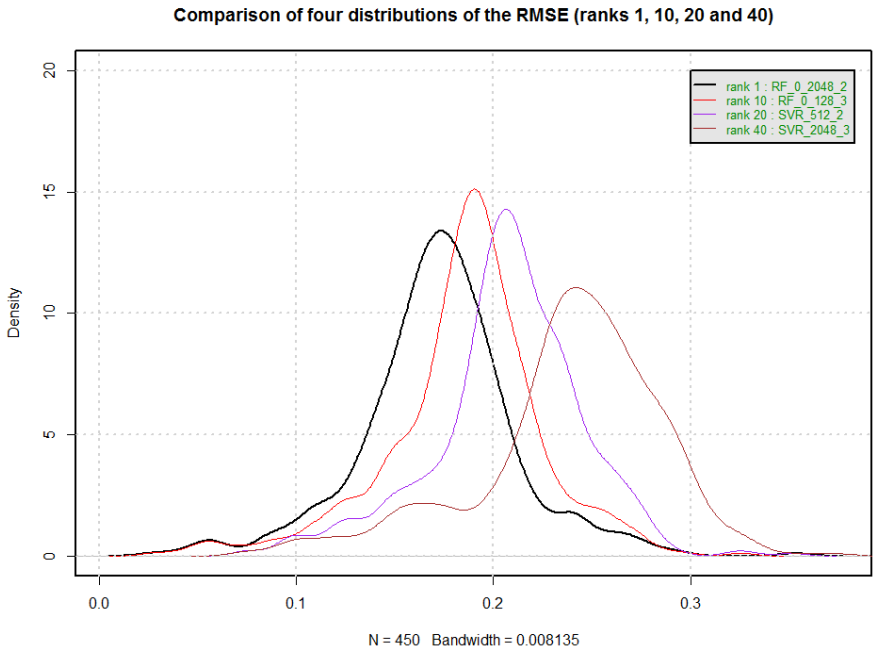

Figure 4.2: Comparison of the distribution of the RMSE for ranks 1, 10, 20 and 40.

To complement the density plot of figure 4.2, a new plot comparing the distribution of RMSE values for 4 different methodologies was made. Following the logic mentioned above, the difference in distributions between the 4 approaches is visible. In agreement with what was seen in table 4.7 the best models are made with RF without maximum depth using ECFP4 and 2048 bits and the worst models were made with SVR using ECFP6 and 2048 bits.

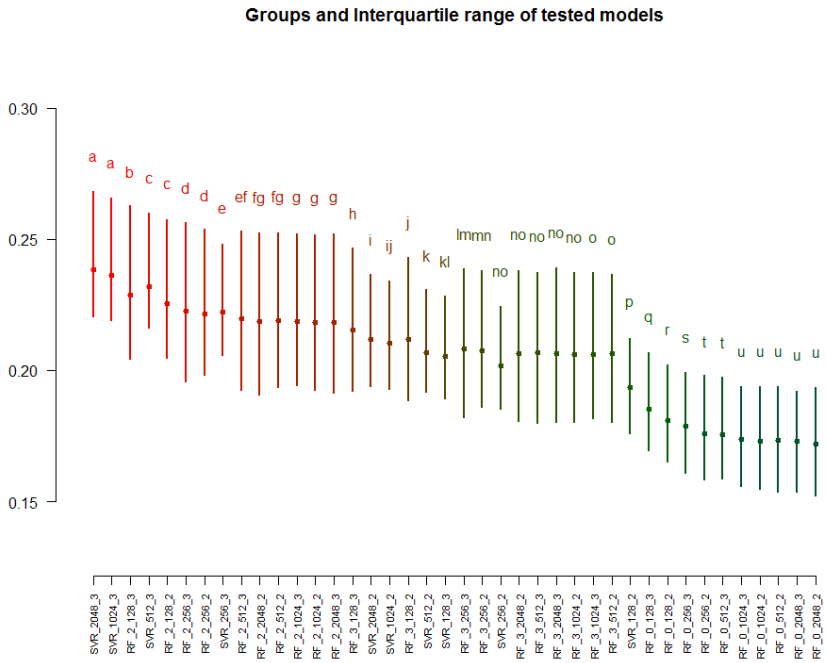

Figure 4.3: Friedman ranking test plot.

To complete the latter information a Friedman plot was made 4.3 where it clearly shows the difference in performance between the 40 approaches used. In the figure, each approach is inserted into a group (represented by a letter) depending on its performance. Approaches within the same group have no statistically significant differences in their performance whereas approaches in different groups have.

In general, it's possible to group the approaches by the type of machine learning algorithm used, with special attention to RF without maximum depth (RF_0). Through the analysis of the graph, it is observed that all 10 approaches using this RF_0 are present in the top 10 methodologies that have the best performing, meaning random forest without depth limitations probably is the best machine learning algorithm to use to build QSAR models.

4.2 Modeling using Feature Selection

With the best set of parameters selected for modeling our data sets the next step was trying to grab these criteria and optimize the process as well as the final result of these. The method used relies heavily upon the weight of each variable in the model. The importance score of each variable is calculated by the RF algorithm and used later by the developed code to create a new model with the help of the SVR algorithm. Doing a control test with only the RF algorithm to create the model resulted in a median for RMSE of 0.1695 and PVE of 0.6101. In the procedure of calculating the importance score of each variable of the model, using only the ones that add something new to the model and creating the model with SVR the median for RMSE was 0.1712 and PVE was 0.5825.

	Mean	**Median**
NVAR	110	110
RMSE before FS	0.1703	0.1695
PVE before FS	0.5661	0.6101
RMSE after FS	0.1749	0.1712
PVE after FS	0.5282	0.5825

Table 4.8: RMSE and PVE before and after using feature selection.

When comparing the values obtained after validation of models created with feature selection (table 4.8 and models created without this technique table 4.7), it is concluded that there are no differences in performance between the approaches.

This is also showed in table 4.8 when comparing the values for RMSE and PVE before and after doing the feature selection. Unfortunately, the results obtained weren't the ones that we expected. Before calculating the importance of the variables 5-fold cross-validation was done with random forest and a model was trained with the training set for the 450 targets, obtaining a median for the RMSE of 0.1695 and a mean for the PVE of 0.6101. After doing the feature selection

and build the final model for the same targets, the median was calculated and was obtained an RMSE of 0.1712 and a PVE of 0.5825. Analyzing these results and comparing them, there is an increase of approximately 1% for the RMSE after feature selection and a decrease of approximately 4% for PVE after feature selection.

However, given the logic of the feature selection technique, the number of variables was reduced compared to other methodologies because many of the variables used in construction of the model didn't add new information to it during the learning phase. Since fewer variables were used to construct models, they become more statistically robust than models created without feature selection as well as tend to generalize better the problems.

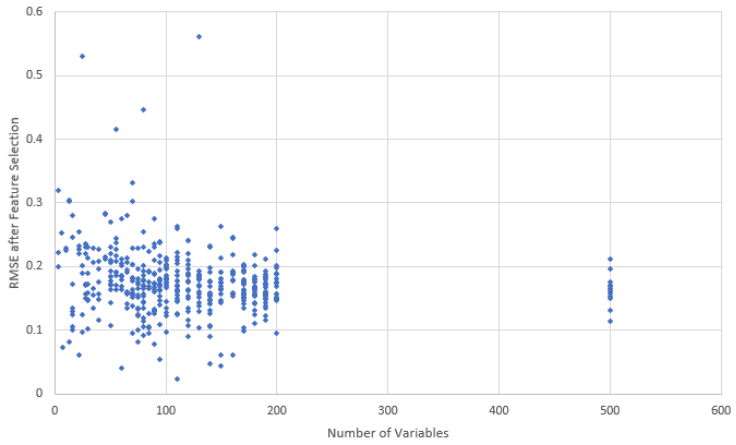

Figure 4.4: Correlation between the number of variables and RMSE after feature selection.

The analysis of scatter plot 4.4, shows that models constructed using a number of variables between 60 and 500, the mean value for RMSE never exceeds 0.19 and the best result is obtained for the number of variables = 75, where RMSE = 0.15 and PVE = 0.59. Note that QSAR models built with more than 150 variables no longer gain prediction power.

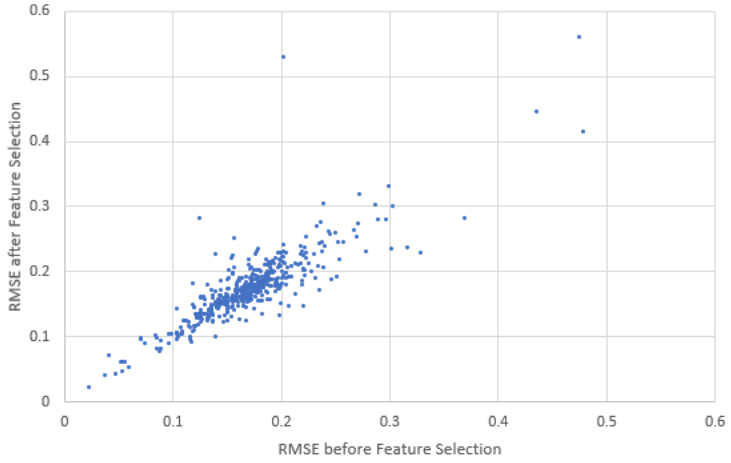

Figure 4.5: Correlation between the values of RMSE before and after feature selection.

When correlating both RMSE's the result can be seen in the scatter plot of figure 4.5. According to the plot it was achieved a high positive correlation between both statistical variables, RMSE before and after feature selection, along with all 450 targets. This is also showed by both the trend line of y = 0.8928X + 0.0229 and the R^2 of 0.7223. This indicates that almost all 450 trained models have positive results in terms of RMSE, except for 6 of them that don't correlate at all. The relationship between both RMSE's leads to the conclusion that QSAR models can only gain by reducing the number of variables, making them more robust and better generalizing the problems.

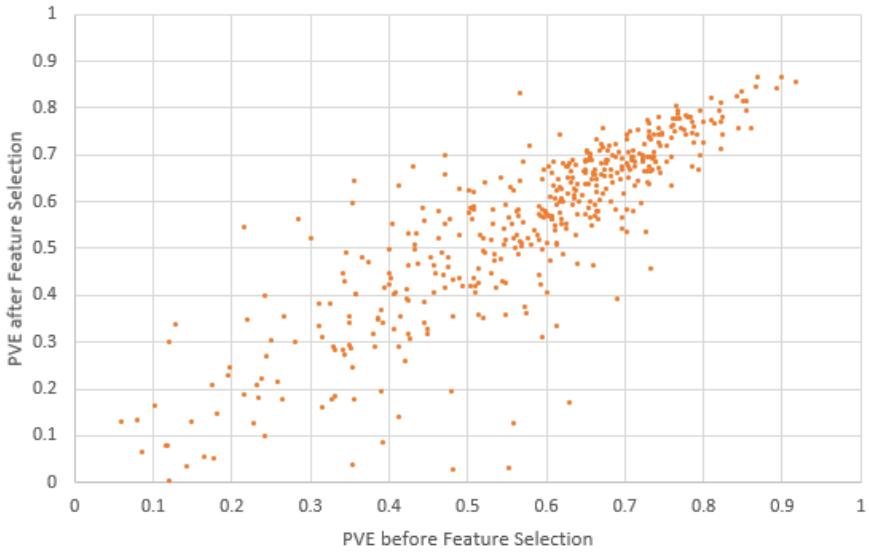

Figure 4.6: Correlation between the values of PVE before and after feature selection.

The figure 4.6 which is the scatter plot for PVEs, its observed that the dots are a lot more dispersed in graphic. This indicates that the the correlation is positive but low. This can be due to the fact of the nature of PVE. Looking to the correlation of the both PVE's for all 450 targets, the result is a trend line of y = 1.0177X + 0.0479 and a R^2 of 0.4606.

Chapter 5

Discussion

5.1 Data set handling result

The initial idea was to work with all the 500 datasets that were extracted from the MIMED database. Unfortunately, it wasn't feasible. At first glance, the idea was to eliminate the duplicates and try to fill missing information by searching it on the online repositories. Almost all the data sets had missing activity values, quantity and even in some cases the SMILES ID. In an initial effort to keep all data inside the data sets search in the ChemBL database were made molecule by molecule but the amount of data missing was very. Given this problem, the decision to eliminate lines with missing information was made, losing a considerable portion of information per data set.

With this problem solved each data set was processed into a new file form, SAR (Structure-Activity Relationship) file, having only the necessary information for the model making, the ID of the molecule, activity between 0 and 1 and the SMILE ID. After this transformation, a basic test was made to see if the data set would produce models given us the second hiccup. A threshold for both RMSE and PVE was set, to guarantee that the QSAR models build were learning and therefore having prediction power. To achieve this a value of 0.30 for RMSE and above 0.30 for PVE was needed for each model. In this test 50 models failed to pass both criteria, meaning that 50 data sets as they were being difficult to model leaving us with 450 data sets to work with the rest of the project. Of this 450 sets around 83% are from the group B of the assay group, meaning that the

majority of the information gathered from online repositories is obtained from binding assays. This is a big bonus for the QSAR modeling due to the nature of the technique itself, which is ligand-based.

5.2 Parameterization of QSAR models results

All possible combinations of machine learning algorithms, Morgan Fingerprints, and the number of bits were made and posteriorly analyzed, to find the approach to make QSAR models for 450 targets.

From this algorithm, 18000 models were made from 450 data sets, meaning that each data set originates 40 models, being equivalent of having, for each machine learning algorithm used, 8 models build corresponding to each of the 5 number of bits used and both fingerprints (4 ML algorithm * 5 bits numbers * 2 ECFP = 40 models per data set). The first type of comparison made was between the 3 assay group to see the firsts differences in the data sets. For this we need to have in mind the differences in the number of data sets for each group, meaning that a group with more data sets has a value more close to the real one. The group A only has 3% of all data sets meaning that the amount of "samples" is far from the ideal, followed by group F with an amount of 14% and finally group B with 83% having the majority of the data sets. In terms of RMSE values, the best overall is group F with approximately 0.16, followed by group A with 0.163 and finally by group B with 0.209. The values shown tell us that the models made by group F have a better chance to predict what is reality. However like said above the number of data sets used in the making of the models has a statistical weight, meaning that probably the group B results are the ones closer to the actual RMSE and PVE values for the given group.

Looking at the results in terms of which algorithm performed better, Random Forest with maximum depth none has a clear lead in both parameters. But does it differ when it comes to a specific configuration of the ML algorithm and number of bits? Models with Random Forest with maximum depth of 2 and 3 were done in the first predictive models to have a means of comparison and to confirm that random forests with deeper trees capture more information about the data when compared with RF with lower depth. Following this logic, only the results for

models made with random forest maximum depth none and models made with support vector regression will be considered.

Grabbing the results for the random forest, when comparing both fingerprints used, the best one to use is ECFP4 with slightly better performance than models made with ECFP6. However, it is clear that the bigger the number of bits, using ECFP4, the better results the model will have, starting at an RMSE of 0.1837 and a PVE of 0.5421 for 128 bits down to an RMSE of 0.1726 and a PVE of 0.5891 when using 2048 bits. This is due to the nature of the algorithm itself combined with the function of the number of bits, which is the higher the number the more information is retained which means, more the model learns and closer it gets to the reality when predicting results. The same cannot be said for SVR results. The bigger the number of bits, using ECFP4, the worst it the performance of models, going from an RMSE of 0.1941 and a PVE of 0.4841 when using 128 bits, to an RMSE of 0.2139 and a PVE of 0.3969 when using 2048 bits. The reason for this may be due to the wrong parameterization of the algorithm or the nature of the algorithm implementation.

Another factor that was taken into account while parameterizing the models was the size of the data sets.

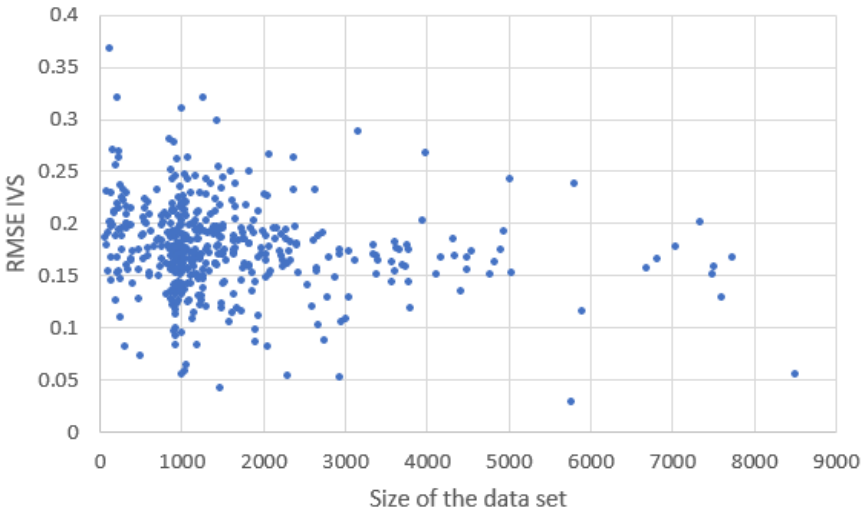

Figure 5.1: Relation between size of the data set with the value of RMSE IVS using ECFP4.

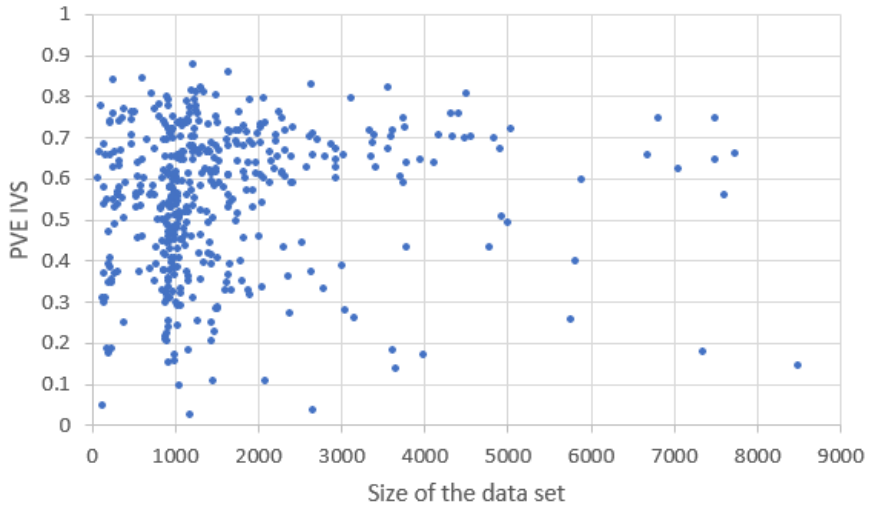

Figure 5.2: Relation between size of the data set with the value of PVE IVS using ECFP4.

Analyses were made to understand what was the impact that the size of the data set could have on the results of training a model (Kausar & Falcao, 2018). As shown in both figure 5.1 and 5.2 at lower sizes the results tend to be non-linear scale between 0.05 up to 0.37 for RMSE and 0.02 to 0.87 for PVE, but as the size of the data grow the results stabilize as shown in the figures. Although this is true, most of our data is centered in size of $<$2000 molecules and because of that results can vary one to another.

With the previous analysis and taking into account all the methodologies used in this work, the best approach for making QSAR models for a large set of problems is using Random Forest without maximum depth, coupled to ECFP4 and number of bits = 2048.

5.3 Modeling using Feacture Selection results

Based on the previous selection of parameters, a method of selection of the most important variables to build a model was performed for all data sets, with the idea of reducing the high dimensionality associated with each problem, making the models more robust and less susceptible to over-fitting.

5.3.1 Comparison of models with and without feature selection

After performing the feature selection the mean value for RMSE is 0.1749 and for PVE is 0.5282. When comparing these values with our previous ones, it's noticeable the impact of feature selection when looking at the results of models built based on support vector regression without feature selection. The decrease in terms of the RMSE is approximately 19% and the increase of the PVE is approximately 41%. Looking and analyzing this values it is safe to conclude that alone the support vector regression algorithm alone is not able to build a good prediction model, but when introduced to feature selection it is able to originate a model with a minimum subset of important variables, that is faster to train, is more robust and have a better prediction capability when compared to the later.

In terms of the models build around the random forest when comparing them versus the models build with feature selection, the results are also positive but with lower impact. The decrease in terms of the RMSE is approximately 1% and the increase of the PVE is approximately 11%. Although the results obtained here aren't as good as the ones obtained versus support vector regression, it still adds a slightest boost to the overall fitness of the models build, as well as the same benefits added to the support vector regression models, a minimum subset of important variables, faster to training of the models, more robustness and have a better prediction capability.

	RMSE	**PVE**
Random Forest Models	0.1769	0.5216
SVR Models	0.2146	0.3448
Feature Selection Models	0.1749	0.5825

Table 5.1: Results for RMSE and PVE for different approaches.

5.4 Kausar Pipeline

To compare the results obtained with the chosen methodology (RF_0), 10 random data sets were chosen and used to obtain results in the workflow developed in KNIME as well as in the approach adopted by this work (Kausar & Falcao, 2018). The Kausar method was developed with the help of R's libraries to enhance the methodology and obtain better results. In addition to this, this later methodology uses the same parameters that the developed method uses except on the fingerprints, which changes twice, using Morgan Fingerprints (Kausar_1) and AtomPairs (Kausar_2). The objective in to understand how the developed methodology performs when compared to another pipeline that generates QSAR models.

To compare the results obtained by the chosen approach, results were obtained using a pipeline developed in KNIME. Ten data sets were randomly chosen from the poll of 450 to build QSAR models. These are used to obtain results in both the chosen methodology of this work and the KNIME approach. To better understanding, the following tables and graphics, all data sets names were changed to the corresponded gene of the ChEMBL ID (table 5.2).

Data Set Names	Gene	Assay Group	Activity	Data Size	Number of Variable used
CHEMBL1941	HRH2	B	IC50	125	30
CHEMBL1827	PDE5A	B	Ki	118	3
CHEMBL286	REN	B	IC50	3953	170
CHEMBL2093865	HDAC3	B	Inhibition	718	50
CHEMBL340	CYP3A4	A	Activity	553	55
CHEMBL210	ADRB2	F	EC50	1102	55
CHEMBL1741189	GFER	B	AC50	1465	170
CHEMBL3687	ALOX12	F	Potency	2674	90
CHEMBL4158	FASN	F	IC50	2060	140
CHEMBL2954	CTSS	B	IC50	1830	180

Table 5.2: Decoding data set name to the corresponding gene name for all 10 data sets chosen.

As previously stated the objective of this step is to compare directly the fitness of models created with the chosen methodology, with the fitness of models created with another pipeline. To do so, the KNIME pipeline was checked and parameters changed to ensure that the general workflow of our methodology was respected.

In the RF_0 methodology, chosen in this work, when performing a 5-fold cross-validation of the random forest algorithm, with Morgan fingerprints (ECFP4) and 2048 bits, coupled with feature selection using support vector regression method with the 10 data sets, the median of the RMSE obtained is 0.19 and the mean of the PVE is 0.48 (Table 5.3).

Gene Names	**RF_0**	
	MF	
	RMSE	**PVE**
HRH2	0.1283	-0.265
PDE5A	0.2304	0.6369
REN	0.2028	0.6444
HDAC3	0.2801	0.3296
CYP3A4	0.2081	0.6798
ADRB2	0.2034	0.7082
GFER	0.1873	0.043
ALOX12	0.1066	-0.057
FASN	0.0972	0.2386
CTSS	0.1787	0.6698
Mean	0.1823	0.3628

Table 5.3: Results for RMSE and PVE using the first methodology.

In the Kausar methodology (Kausar & Falcao, 2018), the majority of the parameters used are equal to the RF_0 methodology, the number of bits (2048) and the feature selection method, to the exception of fingerprints. Two types of fingerprints were used on the KNIME pipeline, the ECFP4 (Kausar_1), and the AtomPair fingerprints (Kausar_2). In Kausar_1, when performing a 5-fold cross-validation of the random forest algorithm, with Morgan fingerprints (ECFP4), coupled with feature selection using support vector regression method using the

10 data sets, the median of the RMSE obtained is 0.18 and the mean of the PVE is 0.52 (Table 5.4). In Kausar_2, when performing 5-fold cross-validation of the random forest algorithm, with AtomPairs, and coupled with feature selection using support vector regression method using the 10 data sets, the median of the RMSE obtained is 0.19 and the mean of the PVE is 0.5 (Table 5.4).

Gene Names	**Kausar Methodology**			
	MF		**AP**	
	RMSE	**PVE**	**RMSE**	**PVE**
HRH2	0.1464	0.2936	0.1756	-0.9081
PDE5A	0.1560	0.7979	0.2046	0.7149
REN	0.1925	0.6363	0.2111	0.5637
HDAC3	0.2513	0.3204	0.2581	0.2749
CYP3A4	0.2118	0.6631	0.2364	0.6155
ADRB2	0.2291	0.6535	0.2222	0.6580
GFER	0.1690	0.0066	0.1759	0.0045
ALOX12	0.0863	0.0642	0.0744	0.0564
FASN	0.0693	0.4082	0.0674	0.4345
CTSS	0.1818	0.6613	0.1816	0.6485
Mean	0.16936	0.45051	0.18072	0.306261

Table 5.4: Results for RMSE and PVE using the second methodology.

Comparing the results of both approaches, it is noteworthy that 3 of the 10 randomly chosen models are difficult to model because of the PVE values being obtained. Performing the Friedman ranking test concludes that there are no significant statistical differences to affirm that one of the three approaches compared stands out in relation to the others. However, it is noteworthy that approach 2 using ECFP4 generally gives the best results.

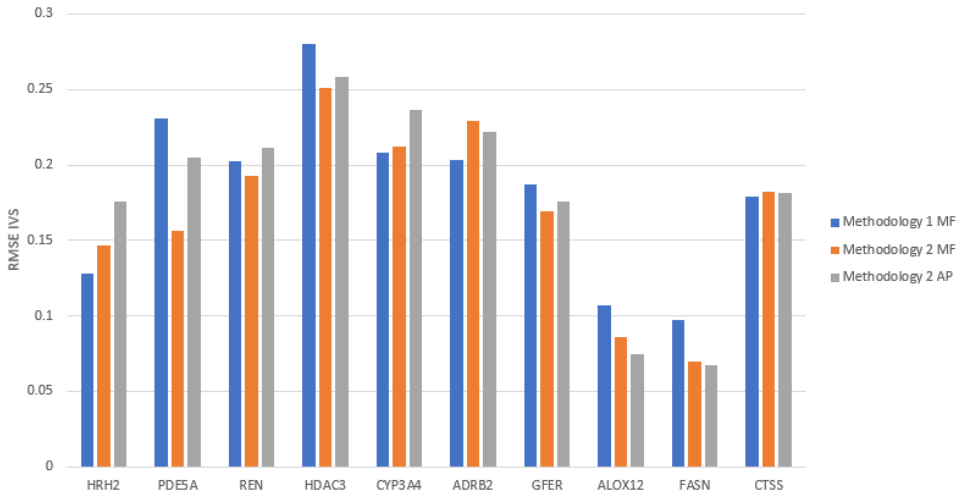

Figure 5.3: Comparison of 10 data sets constructed using different methodologies in terms of RMSE.

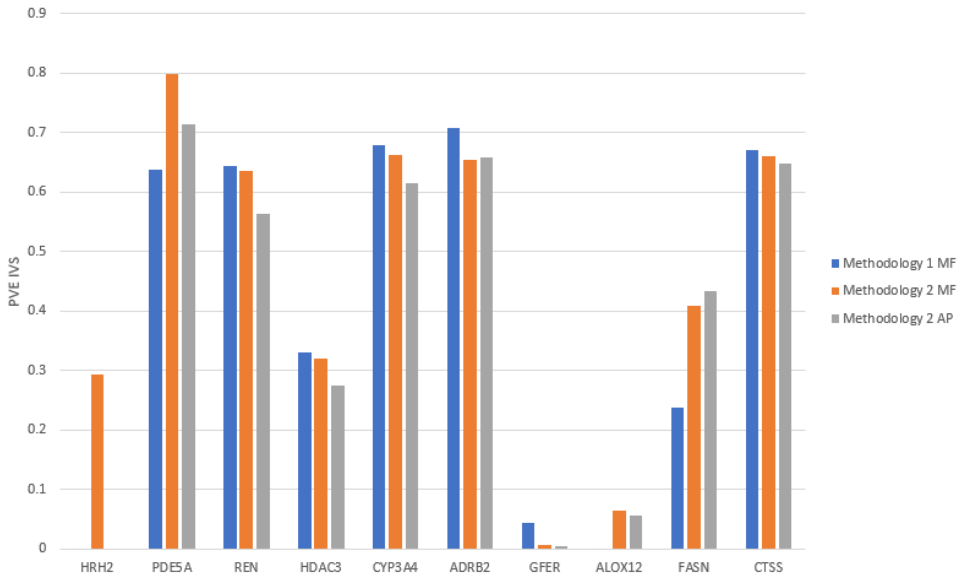

Figure 5.4: Comparison of 10 data sets constructed using different methodologies in terms of PVE.

Looking into the graphics there are differences in the results both in RMSE and PVE (figure 5.3 and figure 5.4). Using the RF_0 methodology, a median for the results of the 10 data sets was calculated resulting in an RMSE of 0.19

and a PVE of 0.48. In terms of Kausar methodology developed in KNIME with Morgan Fingerprints, the median for the RMSE was 0.18 and a PVE of 0.52 were obtained. As for Kausar_2, the method developed in KNIME using AtomPair, the median for the RMSE was 0.19 and a PVE of 0.5 were obtained.

Based only on the results of 10 data sets the best way to make predictive models is to use the methodology developed in KNIME using Morgan Fingerprints. However, one must have in mind that this method was developed using different resources and libraries and with a different purpose. When all 3 methods are analyzed through the Friedman test, it concluded that all 3 do not have any significant statistical differences between then, meaning that depending on the purpose of the task one may choose one method over the other, without having a significant impact when modulating the target.

Chapter 6

Conclusions

In this work, diverse methodologies involving QSAR modeling, with the support of the sklearn machine learning algorithm's library were tested to find the best one to build models for a large set of problems, to help the molecular research environment to predict the pharmacology properties of several molecular targets.

Using the information contained in the MIMED project's database, 500 targets were extracted from it. The data inside those data sets come from different laboratory experiments, meaning that their nature is not the same from one to another. Using the MIMED data sets as a starting point, a script was made to normalize and fix the missing information inside the targets. As a result of this action, information was lost in the normalization of the targets as expected, reducing the size of the data sets up to 49%.

With the data sets normalized the first models could be made. Evaluation norms were specified to ensure that the models were learning from our data. In the first rounds of training and testing the predictive models, was concluded that some targets are very difficult for modeling due to their nature. Because of this problem the poll was shortened in about 50 data sets coming to a total of 450 data sets.

During this procedure, different approaches represented by the combinations of machine learning algorithms, fingerprints and number of bits were tested resulting in a lot of data, which after analyzing it, culminating in the best overall approach to build QSAR models. The best methodology to build predictive models for a large number of problems is using random forest without maximum

depth, ECFP4, and 2048 bits. Although the best set of parameters where chosen it was known that the script could be enhanced to produce better results in terms of building forecasting models.

Using the technique of feature selection for QSAR, random forests were used to calculate the best subset of variables using their importance for model training. Then variables that added more variance to the model were used to build the final model using support vector regression. The final models build represented a better performance in RMSE, from 1% up to 19%, and PVE, from 11% up to 41%.

The methodology was then compared to others developed previously. The methodology chosen to be compared to was developed in KNIME using Java and R. In terms of time spent modeling the problems, the methodology chosen in this work is faster compared to the one developed in KNIME, although the final results in this second methodology appear to have better performance. A Friedman test was made and statistically speaking there are no significant differences between then.

In short, through this work, we have established the best approach to build robust QSAR models in the shortest possible time from a pool of 40 methodologies.

6.1 Future Work

The chosen of this methodology opens many doors in QSAR modeling. The first task to be tested with the work developed in this thesis is to proceed with a functional test of a specific problem. Using the chosen approach, build QSAR models that after doing virtual screening to select molecules with the potential to solve a problem, test them *in vitro*.

The approach chosen, like other, have its flaws. An example of this is the 50 targets that have been set aside. It would be interesting to incorporate new techniques, different machine learning algorithms together with different fingerprints, as well as improvements to this methodology to encompass more complicated modeling problems.

It would also be interesting to see how does this methodology performs comparing to other approaches not worked in this thesis such as distance based methods.

References

Abrantes, P. (2014). Prospeção do espaço métrico molecular para desenho de novos fármacos. implementação de um sistema de informação para moléculas. *Relatório Técnico do Projeto MIMED, Departamento de Informática da Universidade de Lisboa*.

Anderson, A. (2003). The process of structure-based drug design. *Chemistry & Biology*, **10**, 787–797.

ASIKAINEN, A.H., RUUSKANEN, J. & TUPPURAINEN, K.A. (2005). Alternative QSAR models for selected estradiol and cytochrome p450 ligands: comparison between classical, spectroscopic, comfa and grid/golpe methods. *SAR and QSAR in Environmental Research*, **16**, 555–565.

Baroni, M., Costantino, G., Cruciani, G., Riganelli, D., Valigi, R. & Clementi, S. (1993). Generating optimal linear pls estimations (golpe): An advanced chemometric tool for handling 3D-QSAR problems. *Quant. Struct.-Act. Relat.*, **12**, 9–20.

Basak, D. & Patranabis, S.P.D.C. (2007). Support vector regression. *Neural Information Processing - Letters and Reviews*, **11**, 203–224.

Berger, J.R., Choi, D., Kaminski, H.J., Gordon, M.F., Hurko, O., D'Cruz, O., Pleasure, S.J. & Feldman, E.L. (2013). Importance and hurdles to drug discovery for neurological disease. *Annals of Neurology*, **74**, 441–446.

Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K. & Wiswedel, B. (2007). KNIME:

The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*.

Camilo, C., Braga, R. & Andrade, C. (2014). 3D-QSAR approaches in drug design: Perspectives to generate reliable comfa models. *Current Computer-Aided Drug Design*, **10**, 148–159.

Cox, R., Green, D.V.S., Luscombe, C.N., Malcolm, N. & Pickett, S.D. (2013). QSAR workbench: automating qsar modeling to drive compound design. *Future Medical Chemistry*, **27**, 321–336.

Cramer, R.D., Patterson, D.E. & Bunce, J.D. (1988). Comparative molecular field analysis (comfa). 1. effect of shape on binding of steroids to carrier proteins. *J. American Chemical Society*, **110**, 5959–5967.

Dixon, S.L., Duan, J., Smith, E., Bargen, C.D.V., Sherman, W. & Repasky, M.P. (2013). Autoqsar: an automated machine learning tool for best-practice quantitative structure–activity relationship modeling. *Future Medical Chemistry*, **8**, 1825–1839.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, **7**, 1–26.

The drug development process. `https://www.fda.gov/patients/learn-about-drug-and-device-approvals/drug-development-process`, [Online; accessed 2019-05-09].

Freedman, D.A. (2009). *Statistical Models Theory and Practice*.

Gaulton, A., Hersey, A., ł Nowotka, M., Bento, A.P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L.J., Cibrian-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magarinos, M.P., Overington, J.P., Papadatos, G., Smit, I. & Leach, A.R. (2017). The chembl database in 2017. *Nucleic Acids Research*, **45**, 945–954.

Genuer, R., Poggi, J.M. & Tuleau, C. (2008). Random forests: some methodological insights. *Institut National de Recherche en Informatique et en Automatique*.

Genuer, R., Poggi, J.M. & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, **31**, 2225–2236.

Gortari, E., García-Jacas, C., Martinez-Mayorga, K. & Medina-Franco, J. (2017). Database fingerprint (dfp): an approach to represent molecular databases. *Journal of Cheminformatics*, **9**, 1–9.

Grinter, S. & Zou, X. (2014). Challenges, applications, and recent advances of protein-ligand docking in structure-based drug design. *Molecules*, **19**, 10150–10176.

Grisoni, F., Consonni, V. & Todeschini, R. (2018). Impact of molecular descriptors on computational models. *Methods in Molecular Biology*, **1825**.

Hastie (2008). The elements of statistical learning, data mining, inference and prediction. *Springer Series In Statistics*, 745.

Hollander, M., Wolfe, D. & Chicken, E. (2015). Nonparametric statistical methods, third edition. *Wiley Series in Probability and Statistics*.

Houston, D. & Walkinshaw, M. (2012). Consensus docking: Improving the reliability of docking in a virtual screening context. *Journal of Chemical Information And Modeling*, **53**, 384–390.

Kausar, S. & Falcao, A. (2019). Analysis and comparison of vector space and metric space representations in QSAR modeling. *Molecules*, **1698**.

Kausar, S. & Falcao, A.O. (2018). An automated framework for QSAR model building. *Journal of Cheminformatics*, **10**, 1–23.

Koul, A., Becchio, C. & Cavallo, A. (2018). Cross-validation approaches for replicability in psychology. *Frontiers in Psychology*, **9**, 1117.

Rdkit: Open-source cheminformatics. `http://www.rdkit.org`, [Online; accessed 2019-09-15].

REFERENCES

Lionta, E., Spyrou, G., Vassilatis, D.K. & Cournia, Z. (2014). Structure-based virtual screening for drug discovery: Principles, applications and recent advances. *Current Topics in Medicinal Chemistry*, **14**, 1923–1938.

Macalino, S., Gosu, V., Hong, S. & Choi, S. (2015). Role of computer-aided drug design in modern drug discovery. *Archives of Pharmacal Research*, **38**, 1686–1701.

Martins, I.F., Teixeira, A.L., Pinheiro, L. & Falcão, A.O. (2012). Bayesian approach to in silico blood-brain barrier penetration. *Journal of Chemical Information and Modeling*, **52**, 1686–1697.

Morgan, H.L. (1965). The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chem. Doc.*, **2**, 107–113.

Nantasenamat, C., Isarankura-Na-Ayudhya, C. & Prachayasittikul, V. (2010). Advances in computational methods to predict the biological activity of compounds. *Expert. Opin. Drug Discov.*, **5**, 633–654.

Oprea, T. (2002). On the information content of 2d and 3d descriptors for QSAR. *Journal of the Brazilian Chemical Society*, **13**, 811–815.

Drug development and clinical trials. `http://ontheknow.com/new-drugs-clinical-trials/`, [Online; accessed on 2019-10-25].

Rogers, D. & Hahn, M. (2010). Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, **5**, 742–754.

Schneider, G. & Böhm, H.J. (2002). Virtual screening and fast automated docking methods. *Drug Discovery Today*, **7**, 64–70.

Schrider, D. (2018). Supervised machine learning for population genetics: A new paradigm. *Trends in Genetics*, **34**, 301–312.

Smyth, G. (2002). Nonlinear regression. *John Wiley Sons, Ltd, Chichester*, **3**, 1405–1411.

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P. & Feuston, B.P. (2003). Random forest: A classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.*, **43**, 1947–1958.

Teixeira, A., Leal, J. & Falcao, A. (2013). Random forests for feature selection in qspr models - an application for predicting standard enthalpy of formation of hydrocarbons. *Journal of Cheminformatics*, **5**.

Tropsha, A. (2010). Best practices for QSAR model development, validation and exploitation. *Molecular Informatics*.

Vuorinen, A. & Schuster, D. (2015). Methods for generating and applying pharmacophore models as virtual screening filters and for bioactivity profiling. *Methods*, **71**, 113–134.

Warren, G.L., Andrews, C.W., Capelli, A.M., Clarke, B., LaLonde, J., Lambert, M.H., Lindvall, M., Nevins, N., Semus, S.F., Senger, S., Tedesco, G., Wall, I.D., Woolven, J.M., Peishoff, C.E. & Head, M.S. (2006). A critical assessment of docking programs and scoring functions. *Journal of Medicinal Chemistry*, **49**, 5912–5931.